

# Academic Press Library in Signal Processing

Volume 5

Image and Video Compression and Multimedia

# Academic Press Library in Signal Processing

Volume 5

Image and Video Compression and Multimedia  
Editors

**David R. Bull**

*Bristol Vision Institute, University of Bristol, Bristol, UK*

**Min Wu**

*Department of Electrical and Computer Engineering  
and Institute for Advanced Computer Studies,  
University of Maryland, College Park, USA*

**Rama Chellappa**

*Department of Electrical and Computer Engineering  
and Center for Automation Research,  
University of Maryland,  
College Park, MD, USA*

**Sergios Theodoridis**

*Department of Informatics & Telecommunications,  
University of Athens, Greece*



ELSEVIER

AMSTERDAM • WALTHAM • HEIDELBERG • LONDON  
NEW YORK • OXFORD • PARIS • SAN DIEGO  
SAN FRANCISCO • SYDNEY • TOKYO

Academic Press is an imprint of Elsevier



Academic Press is an imprint of Elsevier  
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK  
225 Wyman Street, Waltham, MA 02451, USA

First edition 2014

Copyright © 2014 Elsevier Ltd. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means electronic, mechanical, photocopying, recording or otherwise without the prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone (+44) (0) 1865 843830; fax (+44) (0) 1865 853333; email: [permissions@elsevier.com](mailto:permissions@elsevier.com). Alternatively you can submit your request online by visiting the Elsevier web site at <http://elsevier.com/locate/permissions>, and selecting Obtaining permission to use Elsevier material.

#### Notice

No responsibility is assumed by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein. Because of rapid advances in the medical sciences, in particular, independent verification of diagnoses and drug dosages should be made.

#### Library of Congress Cataloging in Publication Data

A catalog record for this book is available from the Library of Congress

#### British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN: 978-0-12-420149-1

ISSN: 2351-9819

For information on all Elsevier publications  
visit our website at [www.store.elsevier.com](http://www.store.elsevier.com)

Printed and bound in Poland.

14 15 16 17 10 9 8 7 6 5 4 3 2 1



Working together  
to grow libraries in  
developing countries

[www.elsevier.com](http://www.elsevier.com) • [www.bookaid.org](http://www.bookaid.org)

# Introduction

## Signal Processing at Your Fingertips!

Let us flash back to the 1970s when the editors-in-chief of this e-reference were graduate students. One of the time-honored traditions then was to visit the libraries several times a week to keep track of the latest research findings. After your advisor and teachers, the librarians were your best friends. We visited the engineering and mathematics libraries of our Universities every Friday afternoon and poured over the IEEE Transactions, Annals of Statistics, the Journal of Royal Statistical Society, Biometrika, and other journals so that we could keep track of the recent results published in these journals. Another ritual that was part of these outings was to take sufficient number of coins so that papers of interest could be xeroxed. As there was no Internet, one would often request copies of reprints from authors by mailing postcards and most authors would oblige. Our generation maintained thick folders of hard-copies of papers. Prof. Azriel Rosenfeld (one of RC's mentors) maintained a library of over 30,000 papers going back to the early 1950s!

Another fact to recall is that in the absence of Internet, research results were not so widely disseminated then and even if they were, there was a delay between when the results were published in technologically advanced western countries and when these results were known to scientists in third world countries. For example, till the late 1990s, scientists in US and most countries in Europe had a lead time of at least a year to 18 months since it took that much time for papers to appear in journals after submission. Add to this the time it took for the Transactions to go by surface mails to various libraries in the world. Scientists who lived and worked in the more prosperous countries were aware of the progress in their fields by visiting each other or attending conferences.

Let us race back to 21st century! We live and experience a world which is fast changing with rates unseen before in the human history. The era of Information and Knowledge societies had an impact on all aspects of our social as well as personal lives. In many ways, it has changed the way we experience and understand the world around us; that is, the way we learn. Such a change is much more obvious to the younger generation, which carries much less momentum from the past, compared to us, the older generation. A generation which has grew up in the Internet age, the age of Images and Video games, the age of IPAD and Kindle, the age of the fast exchange of information. These new technologies comprise a part of their "real" world, and Education and Learning can no more ignore this reality. Although many questions are still open for discussions among sociologists, one thing is certain. Electronic publishing and dissemination, embodying new technologies, is here to stay. This is the only way that effective pedagogic tools can be developed and used to assist the learning process from now on. Many kids in the early school or even preschool years have their own IPADs to access information in the Internet. When they grow up to study engineering, science, or medicine or law, we doubt if they ever will visit a library as they would by then expect all information to be available at their fingertips, literally!

Another consequence of this development is the leveling of the playing field. Many institutions in lesser developed countries could not afford to buy the IEEE Transactions and other journals of repute. Even if they did, given the time between submission and publication of papers in journals and the time it took for the Transactions to be sent over surface mails, scientists and engineers in lesser developed countries were behind by two years or so. Also, most libraries did not acquire the proceedings of conferences and so there was a huge gap in the awareness of what was going on in technologically advanced

countries. The lucky few who could visit US and some countries in Europe were able to keep up with the progress in these countries. This has changed. Anyone with an Internet connection can request or download papers from the sites of scientists. Thus there is a leveling of the playing field which will lead to more scientist and engineers being groomed all over the world.

The aim of Online Reference for Signal Processing project is to implement such a vision. We all know that asking any of our students to search for information, the first step for him/her will be to click on the web and possibly in the Wikipedia. This was the inspiration for our project. To develop a site, related to the Signal Processing, where a selected set of reviewed articles will become available at a first “click.” However, these articles are fully refereed and written by experts in the respected topic. Moreover, the authors will have the “luxury” to update their articles regularly, so that to keep up with the advances that take place as time evolves. This will have a double benefit. Such articles, besides the more classical material, will also convey the most recent results providing the students/researchers with up-to-date information. In addition, the authors will have the chance of making their article a more “permanent” source of reference, that keeps up its freshness in spite of the passing time.

The other major advantage is that authors have the chance to provide, alongside their chapters, any multimedia tool in order to clarify concepts as well as to demonstrate more vividly the performance of various methods, in addition to the static figures and tables. Such tools can be updated at the author’s will, building upon previous experience and comments. We do hope that, in future editions, this aspect of this project will be further enriched and strengthened.

In the previously stated context, the Online Reference in Signal Processing provides a revolutionary way of accessing, updating and interacting with online content. In particular, the Online Reference will be a living, highly structured, and searchable peer-reviewed electronic reference in signal/image/video Processing and related applications, using existing books and newly commissioned content, which gives tutorial overviews of the latest technologies and research, key equations, algorithms, applications, standards, code, core principles, and links to key Elsevier journal articles and abstracts of non-Elsevier journals.

The audience of the Online Reference in Signal Processing is intended to include practicing engineers in signal/image processing and applications, researchers, PhD students, post Docs, consultants, and policy makers in governments. In particular, the readers can be benefited in the following needs:

- To learn about new areas outside their own expertise.
- To understand how their area of research is connected to other areas outside their expertise.
- To learn how different areas are interconnected and impact on each other: the need for a “helicopter” perspective that shows the “wood for the trees.”
- To keep up-to-date with new technologies as they develop: what they are about, what is their potential, what are the research issues that need to be resolved, and how can they be used.
- To find the best and most appropriate journal papers and keeping up-to-date with the newest, best papers as they are written.
- To link principles to the new technologies.

The Signal Processing topics have been divided into a number of subtopics, which have also dictated the way the different articles have been compiled together. Each one of the subtopics has been coordinated by an AE (Associate Editor). In particular:

1. Signal Processing Theory (Prof. P. Diniz)
2. Machine Learning (Prof. J. Suykens)
3. DSP for Communications (Prof. N. Sidiropoulos)
4. Radar Signal Processing (Prof. F. Gini)
5. Statistical SP (Prof. A. Zoubir)
6. Array Signal Processing (Prof. M. Viberg)
7. Image Enhancement and Restoration (Prof. H. J. Trussell)
8. Image Analysis and Recognition (Prof. Anuj Srivastava)
9. Video Processing (other than compression), Tracking, Super Resolution, Motion Estimation, etc. (Prof. A. R. Chowdhury)
10. Hardware and Software for Signal Processing Applications (Prof. Ankur Srivastava)
11. Speech Processing/Audio Processing (Prof. P. Naylor)
12. Still Image Compression (Prof. David R. Bull)
13. Video Compression (Prof. David R. Bull)
14. Multimedia (Prof. Min Wu)

We would like to thank all the Associate Editors for all the time and effort in inviting authors as well as coordinating the reviewing process. The Associate Editors have also provided succinct summaries of their areas.

The articles included in the current editions comprise the first phase of the project. In the second phase, besides the updates of the current articles, more articles will be included to further enrich the existing number of topics. Also, we envisage that, in the future editions, besides the scientific articles we are going to be able to include articles of historical value. Signal Processing has now reached an age that its history has to be traced back and written.

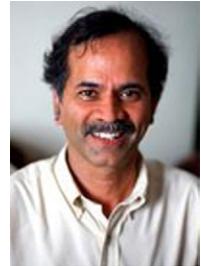
Last but not least, we would like to thank all the authors for their effort to contribute in this new and exciting project. We earnestly hope that in the area of Signal Processing, this reference will help level the playing field by highlighting the research progress made in a timely and accessible manner to anyone who has access to the Internet. With this effort the next breakthrough advances may be coming from all around the world.

The companion site for this work: <http://booksite.elsevier.com/9780124166165> includes multimedia files (Video/Audio) and MATLAB codes for selected chapters.

Rama Chellappa  
Sergios Theodoridis

# About the Editors

**Rama Chellappa** received the B.E. (Hons.) degree in Electronics and Communication Engineering from the University of Madras, India in 1975 and the M.E. (with Distinction) degree from the Indian Institute of Science, Bangalore, India in 1977. He received the M.S.E.E. and Ph.D. Degrees in Electrical Engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively. During 1981–1991, he was a faculty member in the department of EE-Systems at University of Southern California (USC). Since 1991, he has been a Professor of Electrical and Computer Engineering (ECE) and an affiliate Professor of Computer Science at University of Maryland (UMD), College Park. He is also affiliated with the Center for Automation Research, the Institute for Advanced Computer Studies (Permanent Member) and is serving as the Chair of the ECE department. In 2005, he was named a Minta Martin Professor of Engineering. His current research interests are face recognition, clustering and video summarization, 3D modeling from video, image and video-based recognition of objects, events and activities, dictionary-based inference, compressive sensing, domain adaptation and hyper spectral processing.



Prof. Chellappa received an NSF Presidential Young Investigator Award, four IBM Faculty Development Awards, an Excellence in Teaching Award from the School of Engineering at USC, and two paper awards from the International Association of Pattern Recognition (IAPR). He is a recipient of the K.S. Fu Prize from IAPR. He received the Society, Technical Achievement, and Meritorious Service Awards from the IEEE Signal Processing Society. He also received the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At UMD, he was elected as a Distinguished Faculty Research Fellow, as a Distinguished Scholar-Teacher, received an Outstanding Innovator Award from the Office of Technology Commercialization, and an Outstanding GEMSTONE Mentor Award from the Honors College. He received the Outstanding Faculty Research Award and the Poole and Kent Teaching Award for Senior Faculty from the College of Engineering. In 2010, he was recognized as an Outstanding ECE by Purdue University. He is a Fellow of IEEE, IAPR, OSA, and AAAS. He holds four patents.

Prof. Chellappa served as the Editor-in-Chief of IEEE Transactions on Pattern Analysis and Machine Intelligence. He has served as a General and Technical Program Chair for several IEEE international and national conferences and workshops. He is a Golden Core Member of the IEEE Computer Society and served as a Distinguished Lecturer of the IEEE Signal Processing Society. Recently, he completed a two-year term as the President of the IEEE Biometrics Council.

**Sergios Theodoridis** is currently Professor of Signal Processing and Communications in the Department of Informatics and Telecommunications of the University of Athens. His research interests lie in the areas of Adaptive Algorithms and Communications, Machine Learning and Pattern Recognition, Signal Processing for Audio Processing and Retrieval. He is the co-editor of the book "Efficient Algorithms for Signal Processing and System Identification," Prentice Hall 1993, the co-author of the best selling book "Pattern Recognition," Academic Press, 4th ed. 2008, the co-author of the book "Introduction to Pattern Recognition: A MATLAB Approach," Academic Press, 2009, and the co-author of three books in Greek, two of them for the Greek Open University. He is Editor-in-Chief for the Signal Processing Book Series, Academic Press and for the E-Reference Signal Processing, Elsevier.



He is the co-author of six papers that have received best paper awards including the 2009 IEEE Computational Intelligence Society Transactions on Neural Networks Outstanding paper Award. He has served as an IEEE Signal Processing Society Distinguished Lecturer. He was *Otto Monsted Guest Professor*, Technical University of Denmark, 2012, and holder of the *Excellence Chair*, Department of Signal Processing and Communications, University Carlos III, Madrid, Spain, 2011.

He was the General Chairman of EUSIPCO-98, the Technical Program co-Chair for ISCAS-2006 and ISCAS-2013, and co-Chairman and co-Founder of CIP-2008 and co-Chairman of CIP-2010. He has served as President of the European Association for Signal Processing (EURASIP) and as member of the Board of Governors for the IEEE CAS Society. He currently serves as member of the Board of Governors (Member-at-Large) of the IEEE SP Society.

He has served as a member of the Greek National Council for Research and Technology and he was Chairman of the SP advisory committee for the Edinburgh Research Partnership (ERP). He has served as Vice Chairman of the Greek Pedagogical Institute and he was for 4 years member of the Board of Directors of COSMOTE (the Greek mobile phone operating company). He is Fellow of IET, a Corresponding Fellow of the Royal Society of Edinburgh (RSE), a Fellow of EURASIP, and a Fellow of IEEE.

# Section Editors

## Section 1

**David R. Bull** holds the Chair in Signal Processing at the University of Bristol, Bristol, UK. His previous roles include Lecturer with the University of Wales and Systems Engineer with Rolls Royce. He was the Head of the Electrical and Electronic Engineering Department at the University of Bristol, from 2001 to 2006, and is currently the Director of Bristol Vision Institute, a cross-disciplinary organization dedicated to all aspects of vision science and engineering. He is also the Director of the EPSRC Centre for Doctoral Training in Communications. He has worked widely in the fields of image and video processing and video communications and has published some 450 academic papers and articles and has written three books. His current research interests include problems of image and video communication and analysis for wireless, internet, broadcast, and immersive applications. He has been awarded two IET Premiums for this work. He has acted as a consultant for many major companies and organizations across the world, both on research strategy and innovative technologies. He is also regularly invited to advise government and has been a member of DTI Foresight, MoD DSAC, and HEFCE REF committees. He holds many patents, several of which have been exploited commercially. In 2001, he co-founded ProVision Communication Technologies, Ltd., Bristol, and was its Director and Chairman until it was acquired by Global Invacom in 2011. He is a chartered engineer, a Fellow of the IET and a Fellow of the IEEE.



## Section 2

**Min Wu** received the B.E. degree in electrical engineering and the B.A. degree in economics from Tsinghua University, Beijing, China (both with the highest honors), in 1996, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2001. Since 2001, she has been with the University of Maryland, College Park, MD, USA, where she is currently a Professor and a University Distinguished Scholar-Teacher. She leads the Media and Security Team (MAST) at the University of Maryland, with main research interests on information security and forensics and multimedia signal processing. She has published two books and about 145 papers in major international journals and conferences, and holds eight U.S. patents on multimedia security and communications. She is a co-recipient of the two Best Paper Awards from the IEEE Signal Processing Society and EURASIP. She received the NSF CAREER Award in 2002, the TR100 Young Innovator Award from the MIT Technology Review Magazine in 2004, the ONR Young Investigator Award in 2005, the Computer World “40 Under 40” IT Innovator Award in 2007, the IEEE Mac Van Valkenburg Early Career Teaching Award in 2009, and the University of Maryland Invention of the Year Award in 2012. She has served as Vice President – Finance of the IEEE Signal Processing Society from 2010 to 2012, and Chair of the IEEE Technical Committee on Information Forensics and Security from 2012 to 2013. She has been elected an IEEE Fellow for contributions to multimedia security and forensics.



# Authors Biography

## CHAPTER 2

**Béatrice Pesquet-Popescu** received the engineering degree in Telecommunications from the “Politehnica” Institute in Bucharest in 1995 (highest honors) and the Ph.D. thesis from the École Normale Supérieure de Cachan in 1998. In 1998 she was a Research and Teaching Assistant at Université Paris XI and in 1999 she joined Philips Research France, where she worked during two years as a research scientist, then project leader, in scalable video coding. Since October 2000 she is with Télécom ParisTech (formerly, ENST), first as an Associate Professor, and since 2007 as a Full Professor, Head of the Multimedia Group. She is also the Scientific Director of the UBIMEDIA common research laboratory between Alcatel-Lucent Bell Labs and Institut Mines Télécom.



Béatrice Pesquet-Popescu is an IEEE Fellow. In 2013–2014 she serves as a Chair for the Industrial DSC Standing Committee.) and is or was a member of the IVMSP TC, MMSP TC, and IEEE ComSoc TC on Multimedia Communications. In 2008–2009 she was a Member at Large and Secretary of the Executive Subcommittee of the IEEE Signal Processing Society (SPS) Conference Board. She is currently (2012–2013) a member of the IEEE SPS Awards Board. Béatrice Pesquet-Popescu serves as an Editorial Team member for IEEE Signal Processing Magazine, and as an Associate Editor for several other IEEE Transactions.

She holds 23 patents in wavelet-based video coding and has authored more than 260 book chapters, journals, and conference papers in the field. She is a co-editor of the book to appear “Emerging Technologies for 3D Video: Creation, Coding, Transmission, and Rendering,” Wiley Eds., 2013. Her current research interests are in source coding, scalable, robust and distributed video compression, multi-view video, network coding, 3DTV, and sparse representations.

**Marco Cagnazzo** obtained the Laurea (equivalent to the M.S.) degree in Telecommunication Engineering from Federico II University, Napoli, Italy, in 2002, and the Ph.D. degree in Information and Communication Technology from Federico II University and the University of Nice-Sophia Antipolis, Nice, France in 2005. He was a postdoc fellow at I3S Laboratory (Sophia Antipolis, France) from 2006 to 2008. Since February 2008 he has been *Maitre de Conférences* (roughly equivalent to Associate Professor) at Institut Mines-TELECOM, TELECOM ParisTech (Paris), within the Multimedia team. He holds the *Habilitation à Diriger des Recherches* (habilitation) since September 2013. His research interests are content-adapted image coding, scalable, robust, and distributed video coding, 3D and multi-view video coding, multiple description coding, video streaming, and network coding. He is the author of more than 70 scientific contributions (peer-reviewed journal articles, conference papers, book chapters).



Dr. Cagnazzo is an Area Editor for *Elsevier Signal Processing: Image Communication* and *Elsevier Signal Processing*. Moreover he is a regular reviewer for major international scientific reviews (IEEE Trans. Multimedia, IEEE Trans. Image Processing, IEEE Trans. Signal Processing,

IEEE Trans. Circ. Syst. Video Tech., Elsevier Signal Processing, Elsevier Sig. Proc. Image Comm., and others) and conferences (IEEE ICIP, IEEE MMSP, IEEE ICASSP, IEEE ICME, EUSIPCO, and others). He has been in the organizing committees of IEEE MMSP'10, EUSIPCO'12 and he is in the organizing committee of IEEE ICIP'14.

He is an IEEE Senior Member, a Signal Processing Society member and a EURASIP member.

**Frédéric Dufaux** is a CNRS Research Director at Telecom ParisTech. He is also Editor-in-Chief of Signal Processing: Image Communication. He received his M.Sc. in physics and Ph.D. in electrical engineering from EPFL in 1990 and 1994 respectively.

Frédéric has over 20 years of experience in research, previously holding positions at EPFL, Emitall Surveillance, Genimedia, Compaq, Digital Equipment, MIT, and Bell Labs. He has been involved in the standardization of digital video and imaging technologies, participating both in the MPEG and JPEG committees. He is currently co-chairman of JPEG 2000 over wireless (JPWL) and co-chairman of JPSearch. He is the recipient of two ISO awards for these contributions. Frédéric is an elected member of the IEEE Image, Video, and Multidimensional Signal Processing (IVMSP) and Multimedia Signal Processing (MMSP) Technical Committees.

His research interests include image and video coding, distributed video coding, 3D video, high dynamic range imaging, visual quality assessment, video surveillance, privacy protection, image and video analysis, multimedia content search and retrieval, and video transmission over wireless network. He is the author or co-author of more than 100 research publications and holds 17 patents issued or pending.



## CHAPTER 3

**Yanxiang Wang** received the B.S. degree in control system from Hubei University of Technology, China in 2010, and M.Sc. degree in Electrical and Electronic engineering from Loughborough University, UK in 2011. She is currently pursuing the Ph.D degree at Electrical and Electronic Engineering, The University of Sheffield. Her research interests focus on hyper-realistic visual content coding.



**Dr. Charith Abhayaratne** received the B.E. in Electrical and Electronic Engineering from the University of Adelaide in Australia in 1998 and the Ph.D. in Electronic and Electrical Engineering from the University of Bath in 2002. Since 2005, he has been a lecturer within the Department of Electronic and Electrical Engineering at the University of Sheffield in the United Kingdom. He was a recipient of European Research Consortium for Informatics and Mathematics (ERCIM) postdoctoral fellowship in 2002-2003 to carry out research at the Centre for mathematics

and computer science (CWI) in Amsterdam in the Netherlands and at the National Research Institute for computer science and control (INRIA) in Sophia Antipolis in France. From 2004 to 2005, Dr. Abhayaratne was with the Multimedia and Vision laboratory of the Queen Mary, University of London as a senior researcher. Dr. Abhayaratne is the United Kingdom's liaison officer for the European Association for Signal Processing (EURASIP). His research interests include video and image coding, content forensics, multidimensional signal representation, wavelets and signal transforms and visual analysis.



**Marta Mrak** received the Dipl. Ing. and M.Sc. degrees in electronic engineering from the University of Zagreb, Croatia, and the Ph.D. degree from Queen Mary University of London, London, UK. In 2002 she was awarded a German DAAD scholarship and worked on H.264/AVC video coding standard at Heinrich Hertz Institute, Berlin, Germany. From 2003 to 2009, she worked on collaborative research and development projects, funded by the European Commission, while based at the Queen Mary University of London and the University of Surrey (UK). She is currently leading the BBC's research and development project on high efficiency video coding. Her research activities were focused on topics of video coding, scalability, and high-quality visual experience, on which she has published more than 60 papers. She co-edited a book on High-Quality Visual Experience (Springer, 2010) and organized numerous activities on video processing topics, including an IET workshop on "Scalable Coded Media beyond Compression" in 2008 and a special session on "Advances in Transforms for Video Coding" at IEEE ICIP 2011. She is an Elected Member of the IEEE Multimedia Signal Processing Technical Committee and an Area Editor for Elsevier Signal Processing Image Communication journal.



## CHAPTER 4

**Mark Pickering** is an Associate Professor with the School of Engineering and Information Technology, The University of New South Wales, at the Australian Defence Force Academy. Since joining the University of New South Wales, he has lectured in a range of subjects including analog communications techniques, and digital image processing. He has been actively involved in the development of the recent MPEG international standards for audio-visual communications. His research interests include Image Registration, Data Networks, Video and Image Compression, and Error-Resilient Data Transmission.



## CHAPTER 5

**Matteo Naccari** was born in Como, Italy. He received the "Laurea" degree in Computer Engineering (2005) and the Ph.D. in Electrical Engineering and Computer Science (2009)

from Technical University of Milan, Italy. After earning his Ph.D. he spent more than two years as a Postdoc at the Instituto de Telecomunicações in Lisbon, Portugal affiliated with the Multimedia Signal Processing Group. Since September 2011 he joined BBC R&D as Senior Research Engineer working in the video compression team and carrying out activities in the standardization of the HEVC and its related extensions. His research interests are mainly focused in the video coding area where he works or has worked on video transcoding architectures, error resilient video coding, automatic quality monitoring in video content delivery, subjective assessment of video transmitted through noisy channels, integration of human visual system models in video coding architectures, and methodologies to deliver Ultra High Definition (UHD) content in broadcasting applications.



## CHAPTER 6

**Dimitar Doshkov** received the Dipl.-Ing. degree in Telecommunication Engineering from the University of Applied Sciences of Berlin, Germany, in 2008. He joined miControl Parma & Wojcik OHG from 2004 to 2005. He changed in 2006 to SAMSUNG SDI Germany GmbH as a trainee. He has been working for the Fraunhofer Institute for Telecommunications—Heinrich-Hertz-Institut, Berlin, Germany since 2007 and is a Research Associate since 2008. His research interests include image and video processing, as well as computer vision and graphics. He has been involved in several projects focused on image and video synthesis, view synthesis, video coding, and 3D video.



**Patrick Ndjiki-Nya** (M'98) received the Dipl.-Ing. title (corr. to M.S. degree) from the Technische Universität Berlin in 1997. In 2008 he also finished his doctorate at the Technische Universität Berlin. He has developed an efficient method for content-based video coding, which combines signal theory with computer graphics and vision. His approaches are currently being evaluated in equal or similar form by various companies and research institutions in Europe and beyond.



From 1997 to 1998 he was significantly involved in the development of a flight simulation software at Daimler-Benz AG. From 1998 to 2001 he was employed as development engineer at DSPecialists GmbH where he was concerned with the implementation of algorithms for digital signal processors (DSP). During the same period he researched content-based image and video features at the Fraunhofer Heinrich Hertz Institute with the purpose of implementation in DSP solutions from DSPecialists GmbH. Since 2001 he is solely employed at Fraunhofer Heinrich Hertz Institute, where he was Project Manager initially and Senior Project Manager from 2004 on. He has been appointed group manager in 2010.

## CHAPTER 7

**Fan Zhang** works as a Research Assistant in the Visual Information Laboratory, Department of Electrical and Electronic Engineering, University of Bristol, on projects related to parametric video coding and Immersive Technology. Fan received the B.Sc. and M.Sc. degrees from Shanghai Jiao Tong University, Shanghai, China, and his Ph.D. from the University of Bristol. His research interests include perceptual video compression, video metrics, texture synthesis, subjective quality assessment, and HDR formation and compression.



## CHAPTER 8

**Neeraj Gadgil** received the B.E.(Hons.) degree from Birla Institute of Technology and Science (BITS), Pilani, Goa, India, in 2009. He is currently pursuing Ph.D. at School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. Prior to joining Purdue, he worked as a Software Engineer at Cisco Systems (India) Pvt. Ltd., Bangalore, India.

His research interests include image and video processing, video transmission, and signal processing. He is a Graduate Student Member of the IEEE.



**Meilin Yang** received the B.S. degree from Harbin Institute of Technology, Harbin, China, in 2008, and the Ph.D. degree from School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2012.

She joined Qualcomm Inc., San Diego, CA, in 2012, where she is currently a Senior Video System Engineer. Her research interests include image and video compression, video transmission, video analysis, and signal processing.



**Mary Comer** received the B.S.E.E., M.S., and Ph.D. degrees from Purdue University, West Lafayette, Indiana. From 1995 to 2005, she worked at Thomson in Carmel, Indiana, where she developed video processing algorithms for set-top box video decoders. She is currently an Associate Professor in the School of Electrical and Computer Engineering at Purdue University. Her research interests image segmentation, image analysis, video coding, and multimedia systems. Professor Comer has been granted 9 patents related to video coding and processing, and has 11 patents pending. From 2006 to



2010, she was an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology, for which she won an Outstanding Associate Editor Award in 2008. Since 2010, she has been an Associate Editor of the IEEE Transactions on Image Processing. She is currently a member of the IEEE Signal Processing Society Image, Video, and Multidimensional Signal Processing (IVMSP) Technical Committee. She was a Program Chair for the 2009 Picture Coding Symposium (PCS) held in Chicago, Illinois, and also for the 2010 IEEE Southwest Symposium on Image Analysis (SSIAI) in Dallas, Texas. She was the General Chair of SSIAI 2012. She is a Senior Member of IEEE.

**Edward J. Delp** was born in Cincinnati, Ohio. He received the B.S.E.E. (cum laude) and M.S. degrees from the University of Cincinnati, and the Ph.D. degree from Purdue University. From 1980-1984, Dr. Delp was with the Department of Electrical and Computer Engineering at The University of Michigan, Ann Arbor, Michigan. Since August 1984, he has been with the School of Electrical and Computer Engineering and the School of Biomedical Engineering at Purdue University, West Lafayette, Indiana. He is currently the Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering and Professor of Psychological Sciences (Courtesy).

His research interests include image and video compression, medical imaging, multimedia security, multimedia systems, communication, and information theory.

Dr. Delp is a Fellow of the IEEE, a Fellow of the SPIE, a Fellow of the Society for Imaging Science and Technology (IS&T), and a Fellow of the American Institute of Medical and Biological Engineering. In 2004 he received the Technical Achievement Award from the IEEE Signal Processing Society for his work in image and video compression and multimedia security. In 2008 Dr. Delp received the Society Award from IEEE Signal Processing Society (SPS). This is the highest award given by SPS and it cited his work in multimedia security and image and video compression. In 2009 he received the Purdue College of Engineering Faculty Excellence Award for Research. He is a registered Professional Engineer.



## CHAPTER 9

**Dimitris Agrafiotis** is currently Senior Lecturer in Signal Processing at the University of Bristol. He holds an M.Sc. (Distinction) in Electronic Engineering from Cardiff University (1998) and a Ph.D. from the University of Bristol (2002). Dimitris has worked in a number of nationally and internationally funded projects, has published more than 60 papers and holds 2 patents. His work on error resilience and concealment is cited very frequently and has received commendation from, among others, the European Commission. His current research interests include video coding and error resilience, HDR video, video quality metrics, gaze prediction, and perceptual coding.



## CHAPTER 11

**Claudio Greco** received his laurea (B.Eng.) in Computing Engineering in 2004 from the Federico II University of Naples, Italy, his laurea magistrale (M.Eng.) with, honors from the same university in 2007, and his Ph.D. in Signal and Image Processing in 2012, from Télécom ParisTech, France. His research interests include multiple description video coding, multi-view video coding, mobile ad hoc networking, cooperative multimedia streaming, cross-layer optimization for multimedia communications, blind source separation, and network coding.



**Irina Delia Nemoianu** received her engineering degree in Electronics, Telecommunications, and Information Technology in 2009, from the Politehnica Institute, Bucharest, Romania, and her Ph.D. degree in Signal and Image Processing in 2013, from Télécom ParisTech, France. Her research interests include advanced video services, wireless networking, network coding, and source separation in finite fields.



**Marco Cagnazzo** obtained his Laurea (equivalent to the M.Sc.) degree in Telecommunications Engineering from the Federico II University, Naples, Italy, in 2002, and his Ph.D. in Information and Communication Technology from the Federico II University and the University of Nice-Sophia Antipolis, Nice, France in 2005. Since February 2008 he has been Associate Professor at Télécom ParisTech, France with the Multimedia team. His current research interests are scalable, robust, and distributed video coding, 3D and multiview video coding, multiple description coding, network coding, and video delivery over MANETs. He is the author of more than 80 scientific contributions (peer-reviewed journal articles, conference papers, book chapters).



**Jean Le Feuvre** received his Ingénieur (M.Sc.) degree in Telecommunications in 1999, from TELECOM Bretagne. He has been involved in MPEG standardization since 2000 for his NYC-based startup Avipix, llc and joined TELECOM ParisTech in 2005 as Research Engineer within the Signal Processing and Image Department. His main research topics cover multimedia authoring, delivery and rendering systems in broadcast, broadband, and home networking environments. He is the project leader and maintainer of GPAC, a rich media framework based on standard technologies (MPEG, W3C, IETF...). He is the author of many scientific contributions (peer-reviewed journal articles, conference papers, book chapters, patents) in the field and is editor of several ISO standards.



**Frédéric Dufaux** is a CNRS Research Director at Telecom ParisTech. He is also Editor-in-Chief of Signal Processing: Image Communication. He received his M.Sc. in physics and Ph.D. in electrical engineering from EPFL in 1990 and 1994 respectively.

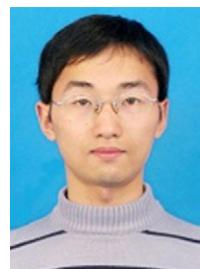
Frédéric has over 20 years of experience in research, previously holding positions at EPFL, Emitall Surveillance, Genimedia, Compaq, Digital Equipment, MIT, and Bell Labs. He has been involved in the standardization of digital video and imaging technologies, participating both in the MPEG and JPEG committees. He is currently co-chairman of JPEG 2000 over wireless (JPWL) and co-chairman of JPSearch. He is the recipient of two ISO awards for these contributions. Frédéric is an elected member of the IEEE Image, Video, and Multidimensional Signal Processing (IVMSP) and Multimedia Signal Processing (MMSP) Technical Committees.



His research interests include image and video coding, distributed video coding, 3D video, high dynamic range imaging, visual quality assessment, video surveillance, privacy protection, image and video analysis, multimedia content search and retrieval, and video transmission over wireless network. He is the author or co-author of more than 100 research publications and holds 17 patents issued or pending.

## CHAPTER 12

**Wengang Zhou** received the B.E. degree in electronic information engineering from Wuhan University, China, in 2006, and the Ph.D. degree in electronic engineering and information science from University of Science and Technology of China, China, in 2011. He was a research intern in Internet Media Group in Microsoft Research Asia from December 2008 to August 2009. From September 2011 to 2013, he works as a postdoc researcher in Computer Science Department in University of Texas at San Antonio. He is currently an associate professor at the Department of Electronic Engineering and Information Science, USTC. His research interest is mainly focused on multimedia content analysis and retrieval.



**Houqiang Li** (S12) received the B.S., M.Eng., and Ph.D. degree from University of Science and Technology of China (USTC) in 1992, 1997, and 2000, respectively, all in electronic engineering. He is currently a professor at the Department of Electronic Engineering and Information Science (EEIS), USTC.

His research interests include multimedia search, image/video analysis, video coding and communication, etc. He has authored or co-authored over 100 papers in journals and conferences. He served as Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology from 2010 to



2013, and has been in the Editorial Board of Journal of Multimedia since 2009. He was the recipient of the Best Paper Award for Visual Communications and Image Processing (VCIP) in 2012, the recipient of the Best Paper Award for International Conference on Internet Multimedia Computing and Service (ICIMCS) in 2012, the recipient of the Best Paper Award for the International Conference on Mobile and Ubiquitous Multimedia from ACM (ACM MUM) in 2011, and a senior author of the Best Student Paper of the 5th International Mobile Multimedia Communications Conference (MobiMedia) in 2009.

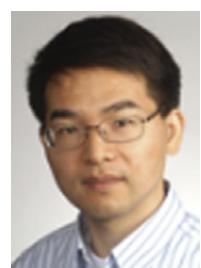
**Qi Tian** (M'96-SM'03) received the B.E. degree in electronic engineering from Tsinghua University, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University in 1996 and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign in 2002. He is currently a Professor in the Department of Computer Science at the University of Texas at San Antonio (UTSA). He took a one-year faculty leave at Microsoft Research Asia (MSRA) during 2008-2009.



Dr. Tian's research interests include multimedia information retrieval and computer vision. He has published over 230 refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA and he also received faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He received the Best Paper Awards in PCM 2013, MMM 2013 and ICIMCS 2012, the Top 10% Paper Award in MMSP 2011, the Best Student Paper in ICASSP 2006, and the Best Paper Candidate in PCM 2007. He received 2010 ACM Service Award. He is the Guest Editors of IEEE Transactions on Multimedia, Journal of Computer Vision and Image Understanding, Pattern Recognition Letter, EURASIP Journal on Advances in Signal Processing, Journal of Visual Communication and Image Representation, and is in the Editorial Board of IEEE Transactions on Multimedia(TMM) and IEEE Transactions on Circuit and Systems for Video Technology (TCSVT), Multimedia Systems Journal, Journal of Multimedia(JMM), and Journal of Machine Visions and Applications (MVA).

## CHAPTER 13

**Zhu Liu** is a Principle Member of Technical Staff at AT&T Labs—Research. He received the B.S. and M.S. degrees in Electronic Engineering from Tsinghua University, Beijing, China, in 1994 and 1996, respectively, and the Ph.D. degree in Electrical Engineering from Polytechnic University, Brooklyn, NY, in 2001. His research interests include multimedia content processing, multimedia databases, video search, and machine learning. He holds 33 US patents and has published more than 60 papers in international conferences and journals. He is on the editorial board of the IEEE Transaction on Multimedia and the Peer-to-peer Networking and Applications Journal.



**Eric Zavesky** joined AT&T Labs Research in October 2009 as a Principle Member of Technical Staff. At AT&T, he has collaborated on several projects to bring alternative query and retrieval representations to multimedia indexing systems including object-based query, biometric representations for personal authentication, and work to incorporate spatio-temporal information into near-duplicate copy detection. His prior work at Columbia University studied semantic visual representations of content and low-latency, high-accuracy interactive search.



**David Gibbon** is Lead Member of Technical Staff in the Video and Multimedia Technologies and Services Research Department at AT&T Labs—Research. His current research focus includes multimedia processing for automated metadata extraction with applications in media and entertainment services including video retrieval and content adaptation. In 2007, David received the AT&T Science and Technology Medal for outstanding technical leadership and innovation in the field of Video and Multimedia Processing and Digital Content Management and in 2001, the AT&T Sparks Award for Video Indexing Technology Commercialization. David contributes to standards efforts through the Metadata Committee of the ATIS IPTV Interoperability Forum. He serves on the Editorial Board for the Journal of Multimedia Tools and Applications and is a member of the ACM, and a senior member of the IEEE. He joined AT&T Bell Labs in 1985 and holds 47 US patents in the areas of multimedia indexing, streaming, and video analysis. He has written a book on video search, several book chapters, and encyclopedia articles as well as numerous technical papers.



**Behzad Shahraray** is the Executive Director of Video and Multimedia Technologies Research at AT&T Labs. In this role, he leads an effort aimed at creating advanced media processing technologies and novel multimedia communications service concepts. He received the M.S. degree in Electrical Engineering, M.S. degree in Computer, Information, and Control Engineering, and Ph.D. degree in Electrical Engineering from the University of Michigan, Ann Arbor. He joined AT&T Bell Laboratories in 1985 and AT&T Labs Research in 1996. His research in multimedia processing has been in the areas of multimedia indexing, multimedia data mining, content-based sampling of video, content personalization and automated repurposing, and authoring of searchable and browsable multimedia content. Behzad is the recipient of the AT&T Medal of Science and Technology for his leadership and technical contributions in content-based multimedia searching and browsing. His work has been the subject of numerous technical publications. Behzad holds 42 US patents in the areas of image, video, and multimedia processing. He is a Senior Member of IEEE, a member of the Association for Computing Machinery (ACM), and is on the editorial board of the International Journal of Multimedia Tools and Applications.



# An Introduction to Video Coding

# 1

**David R. Bull**

*Bristol Vision Institute, University of Bristol, Bristol BS8 1UB, UK*

---

## Nomenclature

1-D	one dimensional
2-D	two dimensional
3-D	three dimensional
<b>AC</b>	alternating current. Used to denote all transform coefficients except the zero frequency coefficient
<b>ADSL</b>	<b>asymmetric digital subscriber line</b>
<b>ASP</b>	advanced simple profile (of MPEG-4)
<b>AVC</b>	advanced video codec (H.264)
B	bi-coded picture
<b>bpp</b>	<b>bits</b> per pixel
bps	bits per second
CCIR	international radio consultative committee (now ITU)
<b>CIF</b>	<b>common intermediate format</b>
codec	encoder and decoder
<b>CT</b>	computerized tomography
CTU	coding tree unit
CU	coding unit
<b>DC</b>	direct current. Refers to zero frequency transform coefficient.
DCT	discrete cosine transform
<b>DFD</b>	displaced frame difference
DFT	discrete Fourier transform
<b>DPCM</b>	<b>differential pulse code modulation</b>
<b>DVB</b>	digital video broadcasting
EBU	European Broadcasting Union

<b>FD</b>	frame difference
fps	frames per second
<b>GOB</b>	group of blocks
GOP	group of pictures
<b>HDTV</b>	high definition television
HEVC	high efficiency video codec (H.265)
HVS	human visual system
I	intra coded picture
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
<b>IP</b>	internet protocol
<b>ISDN</b>	integrated services digital network
ISO	International Standards Organization
ITU	International Telecommunications Union. -R Radio; -T Telecommunications
JPEG	Joint Photographic Experts Group
kbps	kilobits per second
<b>LTE</b>	long term evolution (4G mobile radio technology)
MB	macroblock
mbps	mega bits per second
MC	motion compensation
MCP	motion compensated prediction
ME	motion estimation
MEC	motion estimation and compensation
MPEG	Motion Picture Experts Group
<b>MRI</b>	magnetic resonance imaging
MV	motion vector
P	predicted picture
PSNR	peak signal to noise ratio
<b>QAM</b>	quadrature amplitude modulation 正交
QCIF	quarter CIF resolution
<b>QPSK</b>	quadrature phase shift keying
RGB	red, green, and blue color primaries
SG	study group (of ITU)
SMPTE	Society of Motion Picture and Television Engineers
TV	television

UHDTV	ultra high definition television
UMTS	universal mobile telecommunications system
VDSL	very high bit rate digital subscriber line
VLC	variable length coding
VLD	variable length decoding
YC <sub>b</sub> C <sub>r</sub>	color coordinate system comprising luminance, Y, and two chrominance channels, C <sub>b</sub> and C <sub>r</sub>

## 5.01.1 Introduction

Visual information is the primary consumer of communications bandwidth across all broadcast, internet, and mobile networks. Users are demanding increased video **quality**, increased quantities of video **content**, more extensive **access**, and better **reliability**. This is creating a major tension between the available capacity per user in the network and the bit rates required to transmit video content at the desired quality. Network operators, content creators, and service providers therefore are all seeking better ways to **transmit the highest quality video at the lowest bit rate**, something that can only be achieved through video compression.

**四个要求**

This chapter provides an introduction to some of the most common image and video compression methods in use today and sets the scene for the rest of the contributions in later chapters. It first explains, in the context of a range of video **applications**, **why** compression is needed and **what** compression ratios are required. It then examines the basic video compression **architecture**, using the ubiquitous hybrid, block-based motion compensated codec. Finally it briefly examines why standards are so important in supporting interoperability.

**VC目的**

This chapter, necessarily only provides an overview of video coding algorithms, and the reader if referred to Ref. [1] for a more comprehensive description of the methods used in today's compression systems.

**文章结构**

## 5.01.2 Applications areas for video coding

By 2020 it is predicted that the number of network-connected devices will reach 1000 times the world's population; there will be 7 trillion connected devices for 7 billion people [2]. Cisco predict [3] that this will result in 1.3 zettabytes of global internet traffic in 2016, with over 80% of this being video traffic. This explosion in video technology and the associated demand for video content are driven by:

- Increased numbers of users with increased expectations of quality and mobility.
- Increased amounts of user generated content available through social networking and download sites.
- The emergence of new ways of working using distributed applications and environments such as the cloud.
- Emerging immersive and interactive entertainment formats for film, television, and streaming.

**需求**

### 5.01.2.1 Markets for video technology

A huge and increasing number of applications rely on video technology. These include:

#### 5.01.2.1.1 Consumer video

Entertainment, personal communications, and social interaction provide the primary applications in consumer video, and these will dominate the video landscape of the future. There has, for example, been a massive increase in the consumption and sharing of content on mobile devices and this is likely to be the major driver over the coming years. The key drivers in this sector are:

- Broadcast television, digital cinema and the demand for more immersive content (3-D, multiview, higher resolution, frame rate, and dynamic range).
- Internet streaming, peer to peer distribution, and personal mobile communication systems.
- Social networking, user-generated content, and content-based search and retrieval.
- In-home wireless content distribution systems and gaming.

#### 5.01.2.1.2 Surveillance

We have become increasingly aware of our safety and security, and video monitoring is playing an increasingly important role in this respect. It is estimated that the market for networked cameras (non-consumer) [4] will be \$4.5 billion in 2017. Aligned with this, there will be an even larger growth in video analytics. The key drivers in this sector are:

- Surveillance of public spaces and high profile events.
- National security.
- Battlefield situational awareness, threat detection, classification, and tracking.
- Emergency services, including police, ambulance, and fire.

#### 5.01.2.1.3 Business and automation

Visual communications are playing an increasingly important role in business. For example, the demand for higher quality video conferencing and the sharing of visual content have increased. Similarly in the field of automation, vision-based systems are playing a key role in transportation systems and are now underpinning many manufacturing processes, often demanding the storage or distribution of compressed video content. The drivers in this case can be summarized as:

- Video conferencing, tele-working, and other interactive services.
- Publicity, advertising, news, and journalism.
- Design, modeling, simulation.
- Transport systems, including vehicle guidance, assistance, and protection.
- Automated manufacturing and robotic systems.

#### 5.01.2.1.4 Healthcare

Monitoring the health of the population is becoming increasingly dependent on imaging methods to aid diagnoses. Methods such as CT and MRI produce enormous amounts of data for each scan and these need to be stored as efficiently as possible while retaining the highest quality. Video is also becoming

increasingly important as a point-of-care technology for monitoring patients in their own homes. The primary healthcare drivers for compression are:

- Point-of-care monitoring.
- Emergency services and remote diagnoses.
- Tele-surgery.
- Medical imaging.

It is clear that all of the above application areas require considerable trade-offs to be made between cost, complexity, robustness, and performance. These issues are addressed further in the following section.

### 5.01.3 Requirements of a compression system

#### 5.01.3.1 Requirements

The primary requirement of a video compression system is to produce the highest quality at the lowest bit rate. Other desirable features include:

- 6项要求
- **Robustness to loss:** We want to maintain high quality when signals are transmitted over error-prone channels by ensuring that the bitstream is error-resilient.
  - **Reconfigurability and flexibility:** To support delivery over time-varying channels or heterogeneous networks.
  - **Low complexity:** Particularly for low power portable implementations.
  - **Low delay:** To support interactivity.
  - **Authentication and rights management:** To support conditional access, content ownership verification, or to detect tampering.
  - **Standardization:** To support interoperability.

#### 5.01.3.2 Trade-offs

In practice, it is usual that a compromise must be made in terms of trade-offs between these features because of cost or complexity constraints and because of limited bandwidth or lossy channels. Areas of possible compromise include:

tradeoff

**Lossy vs lossless compression:** We must exploit any redundancy in the image or video signal in such a way that it delivers the desired compression with the minimum perceived distortion. This usually means that the original signal cannot be perfectly reconstructed.

**Rate vs quality:** In order to compromise between bit rate and quality, we must trade off parameters such as frame rate, spatial resolution (luma and chroma), dynamic range, prediction mode, and latency. A codec will include a rate-distortion optimization mechanism that will make coding decisions (for example relating to prediction mode, block size, etc.) based on a rate-distortion objective function [1,5,6].

**Complexity vs cost:** In general, as additional features are incorporated, the video encoder will become more complex. However, more complex architectures invariably are more expensive and may introduce more delay.

**Delay vs performance:** Low latency is important in interactive applications. However, increased performance can often be obtained if greater latency can be tolerated.

**Redundancy vs error resilience:** Conventionally in data transmission applications, channel and source coding have been treated independently, with source compression used to remove picture redundancy and error detection and correction mechanisms added to protect the bitstream against errors. However, in the case of video coding, alternative mechanisms exist for making the compressed bitstream more resilient to errors, or dynamic channel conditions, or for concealing errors at the decoder. Some of these are discussed in Chapters 8 and 9.

### 5.01.3.3 How much do we need to compress?

压缩比

Typical video compression ratio requirements are currently between 100:1 and 200:1. However this could increase to many hundreds or even thousands to one as new more demanding formats emerge.

原始态  
RGB YUV

#### 5.01.3.3.1 Bit rate requirements

Pictures are normally acquired as an array of color samples, usually based on combinations of the red, green and blue primaries. They are then usually converted to some other more convenient color space, such as  $Y, C_b, C_r$  that encodes luminance separately to two color difference signals [1]. Table 1.1 shows typical sampling parameters for a range of common video formats. Without any compression, it can be seen, even for the lower resolution formats, that the bit rate requirements are high—much higher than what is normally provided by today’s communication channels. Note that the chrominance signals are encoded at a reduced resolution as indicated by the 4:2:2 and 4:2:0 labels. Also note that two formats are included for the HDTV case (the same could be done for the other formats); for broadcast

**Table 1.1** Typical Parameters for Common Digital Video Formats and their (Uncompressed) Bit Rate Requirements

Format	Spatial sampling (V × H)	Temporal sampling (fps)	Raw bit rate (30 fps, 8/10 bits)
UHDTV (4:2:0) (ITU-R 2020)	Lum: 7680 × 4320 Chrom: 3840 × 2160	24, 25, 30, 50, 60, 120	14,930 Mbps <sup>a</sup>
HDTV (4:2:0) (ITU-R 709)	Lum: 1920 × 1080 Chrom: 960 × 540	24, 25, 30, 50, 60	933.1 Mbps <sup>a</sup>
HDTV (4:2:2) (ITU-R 709)	Lum: 1920 × 1080 Chrom: 960 × 1080	24, 25, 30, 50, 60	1244.2 Mbps <sup>a</sup>
SDTV (ITU-R 601)	Lum: 720 × 576 Chrom: 360 × 288	25, 30	149.3 Mbps
CIF	Lum: 352 × 288 Chrom: 176 × 144	10–30	36.5 Mbps
QCIF	Lum: 176 × 144 88 × 72	5–30	9.1 Mbps

<sup>a</sup> Encoding at 10 bits.

UHDTV = Ultra High Definition Television; HDTV = High Definition Television; SDTV = Standard Definition Television; CIF = Common Intermediate Format; QCIF = Quarter CIF.

**基本格式**

quality systems, the 4:2:2 format is actually more representative of the original bit rate as this is what is produced by most high quality cameras. The **4:2:0 format**, on the other hand, is that normally employed for transmission after compression.

Finally, it is worth highlighting that the situation is actually worse than that shown in [Table 1.1](#) especially for the new Ultra High Definition (UHDTV) standard [7] where higher frame rates and longer wordlengths will normally be used. For example at 120 frames per second (fps) with a 10 bit wordlength for each sample, the raw bit rate increases to 60 Gbps for a single video stream! This will increase even further if 3-D or multiview formats are employed.

### **5.01.3.3.2 Bandwidth availability**

Let us now examine the bandwidths available in typical communication channels as summarized in [Table 1.2](#). This table shows the theoretical maximum bit rates under optimum operating conditions and it should be noted that these are rarely, if ever, achieved in practice. The bit rates available to an individual user at the application layer will normally be significantly lower than the figures quoted in [Table 1.2](#). The effective throughput is influenced by a large range of internal and external factors including: overheads due to link layer and application layer protocols; network contention, congestion, and numbers of users; asymmetry between download and upload rates; and of course the prevailing channel conditions. In particular, as channel conditions deteriorate, modulation and coding schemes will need to be increasingly robust. This will create lower spectral efficiency with increased coding overhead needed in order to maintain a given quality. The number of retransmissions will also inevitably increase as the channel worsens. As an example, DVB-T2 will reduce from 50 Mbps (256QAM @ 5/6 code-rate) to around 7.5 Mbps when channel conditions dictate a change in modulation and coding mode down to 1/2 rate QPSK. Similarly for 802.11n, realistic bandwidths per user can easily reduce well below 10 Mbps. 3G download speeds never offer 384 kbps—more frequently they will be less than 100 kbps.

Consider the example of a digital HDTV transmission at 30 fps using DVB-T2, where the average bit rate allowed in the multiplex (per channel) is 15 Mbps. The raw bit rate, assuming a 4:2:2 original at 10 bits, is approximately 1.244 Gbps, while the actual bandwidth available dictates a bit rate of 15 Mbps. This represents a compression ratio of approximately 83:1. Download sites such as YouTube typically support up to 6 Mbps for HD 1080p format, but more often video downloads will use 360p or 480p ( $640 \times 480$  pixels) formats at 30 fps, with a bit rate between 0.5 and 1 Mbps encoded using the H.264/AVC [8] standard. In this case the raw bit rate, assuming color subsampling in 4:2:0 format, will be 110.6 Mbps. As we can see, this is between 100 and 200 times the bit rate supported for transmission.

**Table 1.2** Theoretical Bandwidth Characteristics for Common Communication Systems

Communication system	Maximum bandwidth
3G mobile (UMTS)	384 kbps
4G mobile ( $4 \times 4$ LTE)	326 Mbps
Broadband (ADSL2)	24 Mbps
Broadband (VDSL2)	100 Mbps
WiFi (IEEE 802.11n)	600 Mbps
Terrestrial TV (DVB-T2 (8 MHz))	50 Mbps

## 5.01.4 The basics of compression

流程

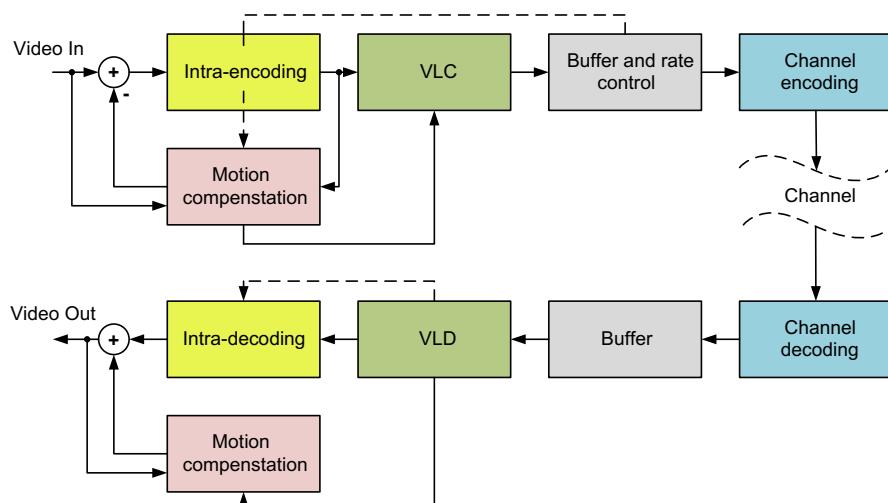
A simplified block diagram of a video compression system is shown in [Figure 1.1](#). This shows an input being encoded, transmitted, and decoded.

### 5.01.4.1 Still image encoding

If we ignore the blocks labeled as motion compensation, the diagram in [Figure 1.1](#) describes a still image encoding system, such as that used in JPEG [9]. The intra-frame encoder performs coding of the picture without reference to any other frames. This is normally achieved by exploiting spatial redundancy through transform-based decorrelation followed by **variable length symbol encoding** (VLC). The image is then conditioned for transmission using some means of error-resilient coding that makes the encoded bitstream more robust to channel errors. At the decoder, the inverse operations are performed and the original image is reconstructed at the output.

### 5.01.4.2 Video encoding

A video signal can be considered as a sequence of still images, acquired typically at a rate of 24, 25, 30, 50, or 60 fps. Although it is possible to encode a video sequence as a series of still images using intra-frame methods as described above, we can achieve significantly higher coding efficiency if we also exploit the temporal redundancy that exists in most natural video sequences. This is achieved using inter-frame motion prediction as represented by the motion compensation block in [Figure 1.1](#). This block predicts the structure of the incoming video frame based on the contents of previously encoded



**FIGURE 1.1**

Simplified high level video compression architecture.

MC  
剩余帧  
buffer  
传输

frames. The encoding continues as for the intra-frame case, except this time the intra-frame encoder block processes the **low energy residual signal** remaining after prediction, rather than the original frame. After variable length encoding, the encoded signal will be buffered prior to transmission. The **buffer** serves to **smooth out content-dependent variations in the output bit rate** and the buffer is managed by a **rate controller algorithm** which **adjusts coding parameters in order to match the video output to the instantaneous capacity of the channel**.

结果：

Because of the reliance on both spatial and temporal prediction, compressed video bitstreams are more **prone to channel errors** than still images, suffering from temporal as well as spatial error propagation. Methods of mitigating this, making the bitstream more robust and correcting or concealing the resulting artifacts are described in later chapters.

基本单位

### 5.01.4.3 Coding units and macroblocks

Video compression algorithms rarely process information at the scale of a picture or a pixel. Instead the coding unit is normally a square block of pixels. In standards up to and including H.264, this took the form of a  **$16 \times 16$**  block, comprising **luma** and **chroma** information, called a **macroblock**.

#### 5.01.4.3.1 Macroblocks

A typical macroblock structure is illustrated in [Figure 1.2](#). The macroblock shown corresponds to what is known as a 4:2:0 format [1] and comprises a  $16 \times 16$  array of luma samples and two subsampled  $8 \times 8$  arrays of chroma (color difference) samples. This macroblock structure, when coded, must include all of the information needed to reconstruct the spatial detail. For example, this might include **transform coefficients**, **motion vectors**, **quantizer information**, and other information relating to further block partitioning for prediction purposes. A  $16 \times 16$  block size is normally the base size used for motion estimation; within this, the decorrelating transforms are normally applied at either  $8 \times 8$  or  $4 \times 4$  levels.

其他信息

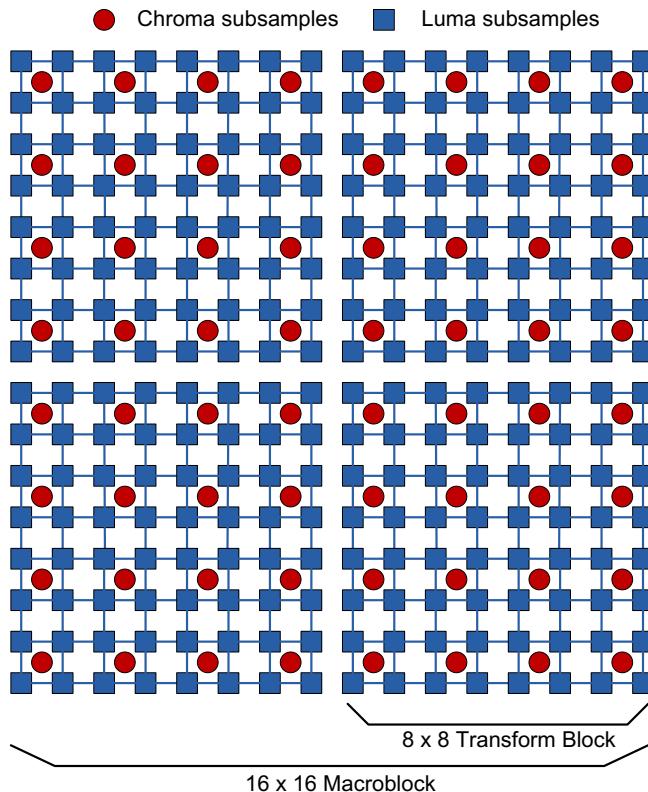
#### 5.01.4.3.2 Coding tree units

The recent HEVC coding standard [10, 11] has extended the size of a macroblock up to  $64 \times 64$  samples to support higher spatial resolutions, with transform sizes up to  $32 \times 32$ . It also provides much more flexibility in terms of block partitioning to support its various prediction modes. Further details on the HEVC standard are provided in [Chapter 3](#).

评价

#### 5.01.4.4 Video quality assessment

The most obvious way of assessing video quality is to ask a human viewer. **Subjective testing** methodologies have therefore become an important component in the design and optimization of new compression systems. However, such tests are costly and time consuming and cannot be used for real-time rate-distortion optimization. Hence **objective metrics** are frequently used instead as these can provide an instantaneous estimate of video quality. These are discussed alongside subjective evaluation methods in [Chapter 7](#). In particular, metrics that can more accurately predict visual quality, aligned with the HVS are highly significant in the context of future coding strategies such as those discussed in [Chapters 5](#) and [6](#).

**FIGURE 1.2**

Typical macroblock structure.

### 5.01.5 Decorrelating transforms

Transformation presents a convenient basis for compression and this comes about through three mechanisms:

1. It provides data decorrelation and creates a frequency-related distribution of energy allowing low energy coefficients to be discarded.
2. Retained coefficients can be quantized, using a scalar quantizer, according to their perceptual importance.
3. The sparse matrix of all remaining quantized coefficients exhibits symbol redundancy which can be exploited using variable length coding.

For the purposes of transform coding, an input image is normally segmented into small  $N \times N$  blocks where the value of  $N$  is chosen to provide a compromise between complexity and decorrelation

大小，  
依据

**performance**. Transformation maps the raw input data into a representation more amenable to compression. Decorrelating transforms, when applied to correlated data, such as natural images, produce energy compaction in the transform domain coefficients and these can be quantized to reduce the dynamic range of the transformed output, according to a fidelity and/or bit rate criterion. For correlated spatial data, the resulting block of coefficients after quantization will be sparse. Quantization is not a reversible process, hence once quantized, the original signal cannot be perfectly reconstructed and some degree of signal distortion is introduced. This is thus the basis of lossy compression.

### 5.01.5.1 The discrete cosine transform (DCT)

The discrete cosine transform was first introduced by Ahmed et al. in 1974 [12] and is the most widely used unitary transform for image and video coding applications. Like the discrete Fourier transform, the DCT provides information about a signal in the **frequency domain**. However, unlike the DFT, the **DCT of a real-valued signal is itself real valued** and importantly it also **does not introduce artifacts due to periodic extension of the input data**.

With the DFT, a finite length data sequence is naturally extended by periodic extension. Discontinuities in the time (or spatial) domain therefore produce ringing or spectral leakage in the frequency domain. This can be avoided if the data sequence is symmetrically (rather than periodically) extended prior to application of the DFT. This produces an even sequence which has the added benefit of yielding real-valued coefficients. The DCT is not as useful as the DFT for frequency domain signal analysis due to its deficiencies when representing pure sinusoidal waveforms. However, in its primary role of signal compression, it performs exceptionally well.

As we will see, the DCT has good **energy compaction properties** and its performance approaches that of the optimum transform for correlated image data. The 1-D DCT, in its most popular form, is given by:

$$c(k) = \sqrt{\frac{2}{N}} \varepsilon_k \sum_{m=0}^{N-1} x[m] \cos\left(\frac{\pi k}{N}\left(m + \frac{1}{2}\right)\right). \quad (1.1)$$

Here  $N$  is the transform dimension,  $c(k)$  are the transform coefficients, and  $x[m]$  are the input data. Similarly the 2-D DCT is given by:

$$c(k, l) = 2 \frac{\varepsilon_k \varepsilon_l}{\sqrt{NM}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] \cos\left(\frac{\pi k}{N}\left(m + \frac{1}{2}\right)\right) \cos\left(\frac{\pi l}{M}\left(n + \frac{1}{2}\right)\right), \quad (1.2)$$

$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0, \\ 1 & \text{otherwise.} \end{cases}$$

The 2-D DCT basis functions are shown for the case of the  $8 \times 8$  DCT in [Figure 1.3](#). Further details on the derivation and characteristics of the DCT can be found in [1].

### 5.01.5.2 Coefficient quantization

Quantization is an important step in lossy compression as it **provides the basis for creating a sparse matrix of quantized coefficients** that can be efficiently entropy coded for transmission. It is however an **irreversible** operation and must be carefully managed—one of the challenges is to perform quantization

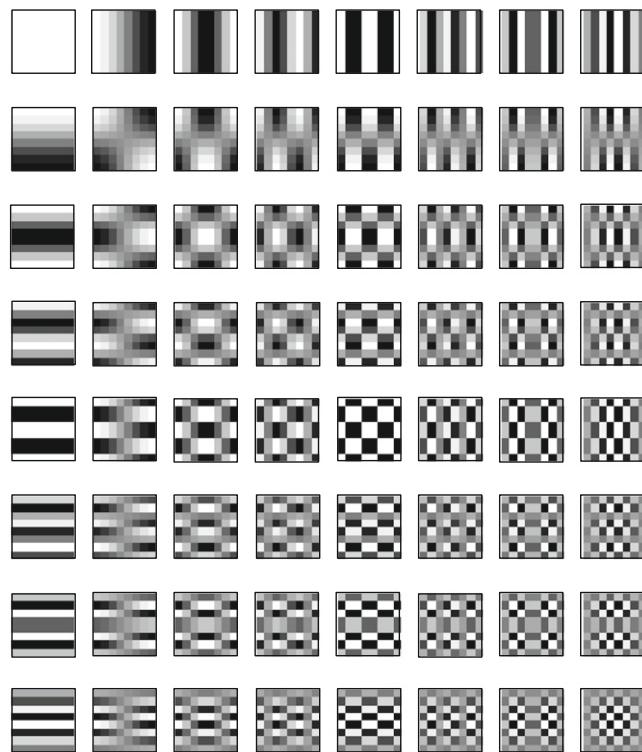
两个优势

DCT, DFT比较

数据维数  
阶数

作用

不可逆

**FIGURE 1.3**

2-D DCT basis functions for  $N = 8$ .

### 过程 , 组成

in such a way as to minimize its psychovisual impact. The quantizer comprises a set of **decision** levels and a set of **reconstruction** levels.

### 场景

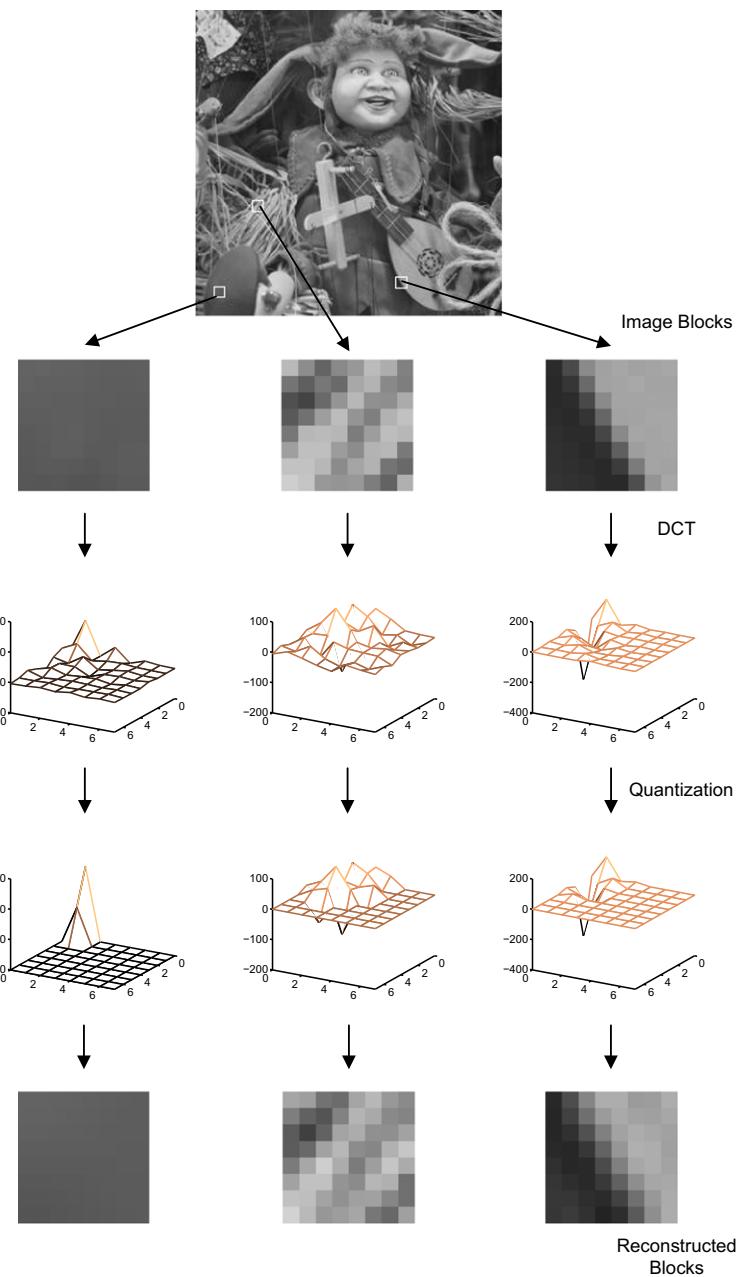
**Intra-frame** transform coefficients are normally quantized using a uniform **quantizer**, with the **coefficients** pre-weighted to reflect the frequency dependent sensitivity of the human visual system. A general expression which captures this is given in [Eq. \(1.3\)](#), where  $Q$  is the quantizer step-size,  $k$  is a constant, and  $\mathbf{W}$  is a coefficient-dependent weighting matrix obtained from psychovisual experiments.

$$c_Q(i, j) = \left\lceil \frac{kc(i, j)}{QW_{i,j}} \right\rceil. \quad (1.3)$$

After transmission or storage, we must rescale the quantized transform coefficients prior to inverse transformation, thus:

$$\tilde{c}(i, j) = \frac{c_Q(i, j)QW_{i,j}}{k}. \quad (1.4)$$

An example of the effects of coefficient quantization on reconstruction quality for a range of block types is shown in [Figure 1.4](#). It can be observed that more textured blocks require larger numbers of

**FIGURE 1.4**

Effects of coefficient quantization on various types of data block.

coefficients in order to create a good approximation to the original content. The best reconstruction can be achieved with fewer coefficients for the case of untextured blocks, as shown for the left-hand block in the figure.

## 5.01.6 Symbol encoding

The sparsity of the quantized coefficient matrix can be exploited (typically by run-length coding) to produce a compact sequence of symbols. The symbol encoder assigns a codeword (a binary string) to each symbol. The code is designed to reduce coding redundancy and it normally uses variable length codewords. This operation is **reversible**.

### 5.01.6.1 Dealing with sparse matrices

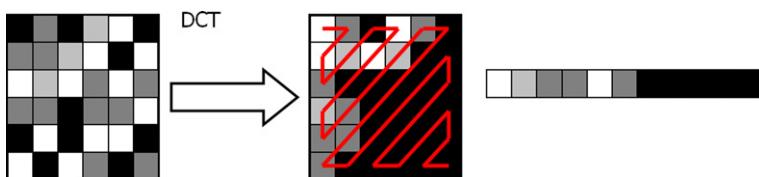
**结果** After applying a forward transform and quantization, the resulting matrix contains a relatively small proportion of non-zero entries with most of its energy compacted toward the lower frequencies (i.e. the top left corner of the matrix). In such cases, **run-length coding (RLC)** can be used to efficiently represent long strings of identical values by grouping them into a single symbol which codes the **value** and the number of **repetitions**. This is a simple and effective method of reducing redundancies in a sequence.

**准备：扫描** In order to perform run-length encoding, we need to convert the 2-D coefficient matrix into a 1-D vector and furthermore we want to do this in such a way that maximizes the runs of zeros. Consider for example the  $6 \times 6$  block of data and its transform coefficients in Figure 1.5. If we scan the matrix using a **zig-zag** pattern, as shown in the figure, then this is more energy-efficient than scanning by rows or columns.

### 5.01.6.2 Entropy encoding

Several methods exist that can exploit data statistics during symbol encoding. The most relevant of these in the context of image and video encoding are:

- **Huffman coding [13]:** This is a method for coding individual symbols at a rate close to the **first-order entropy**, often used in conjunction with other techniques in a lossy codec.
- **Arithmetic coding [14–16]:** This is a more sophisticated method which is capable of achieving fractional bit rates for symbols, thereby providing greater compression efficiency for more common symbols.



**FIGURE 1.5**

Zig-zag scanning prior to variable length coding.

**联合使用** Frequently, the above methods are used in combination. For example, DC DCT coefficients are often encoded using a combination of predictive coding (DPCM) and either Huffman or arithmetic coding. Furthermore, motion vectors are similarly encoded using a form of predictive coding to condition the data prior to entropy coding.

---

## 5.01.7 Motion estimation

For still natural images, significant spatial redundancies exist and we have seen that these can be exploited via decorrelating transforms. The simplest approach to encoding a sequence of moving images is thus to apply an *intra-frame* (still image) coding method to each frame. This can have some benefits, especially in terms of the error resilience properties of the codec. However it generally results in limited compression performance.

For real-time video transmission over low bandwidth channels, there is often insufficient capacity to code each frame in a video sequence independently (25–30 fps is required to avoid flicker). The solution is thus to exploit the temporal correlation that exists between temporally adjacent frames in a video sequence. This *inter-frame* redundancy can be reduced through motion prediction, resulting in further improvements in coding efficiency.

In motion compensated prediction, a **motion model** (usually block-based translation only) is assumed and **motion estimation** (ME) is used to estimate the motion that occurs between the **reference frame** and the **current frame**. Once the motion is estimated, a process known as **motion compensation** (MC) is invoked to use the motion information from ME to modify the contents of the reference frame, according to the motion model, in order to produce a prediction of the current frame. The prediction is called a **motion-compensated prediction** (MCP) or a **displaced frame** (DF). The prediction error is known as the **displaced frame difference** (DFD) signal. Figure 1.6 shows how the pdf of pixel values is modified for FD and DFD frames, compared to an original frame from the *Football* sequence.

A thorough description of motion estimation methods and their performance is provided in Chapter 2 and in [1].

---

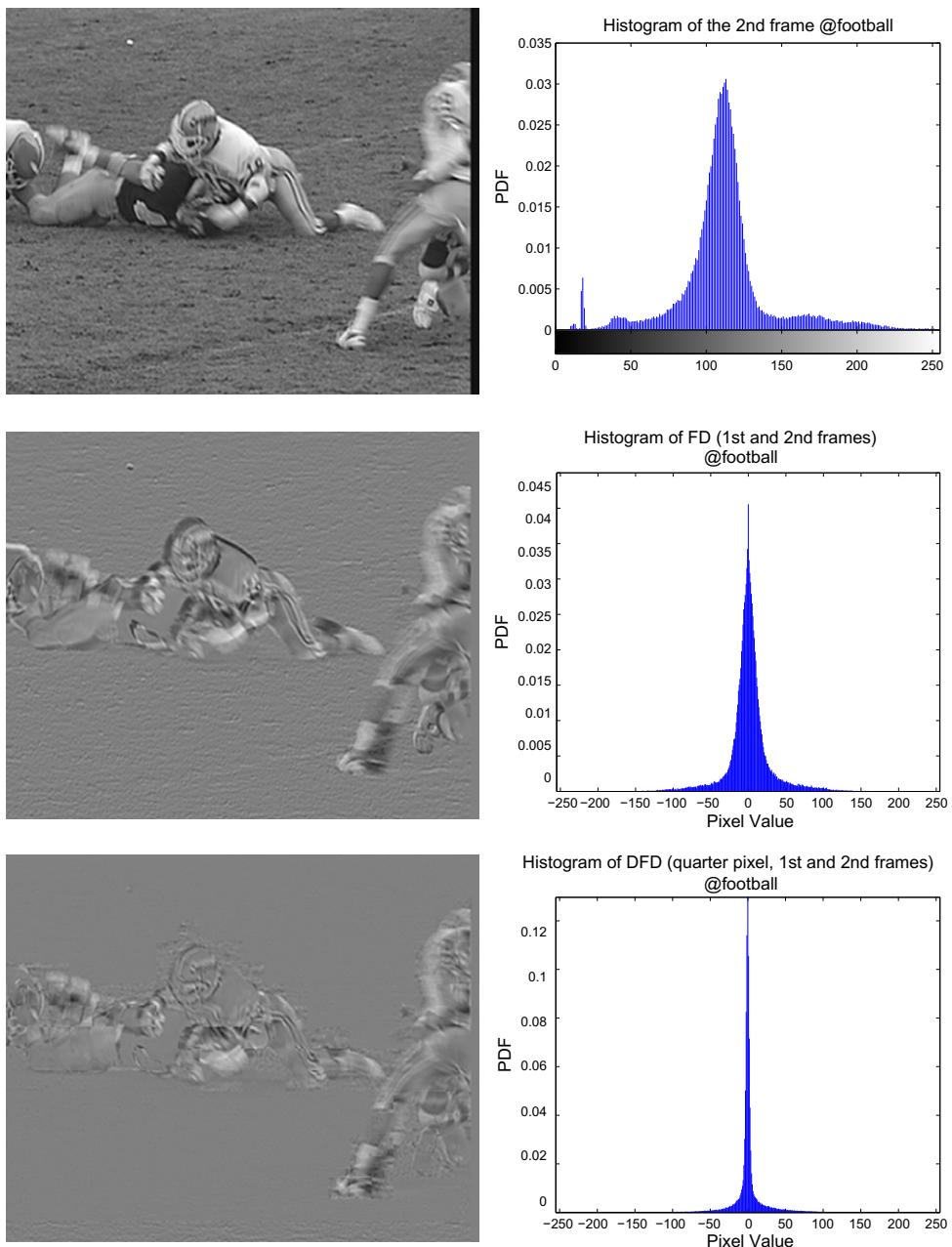
## 5.01.8 The block-based motion-compensated video coding architecture

### 5.01.8.1 Picture types and prediction modes

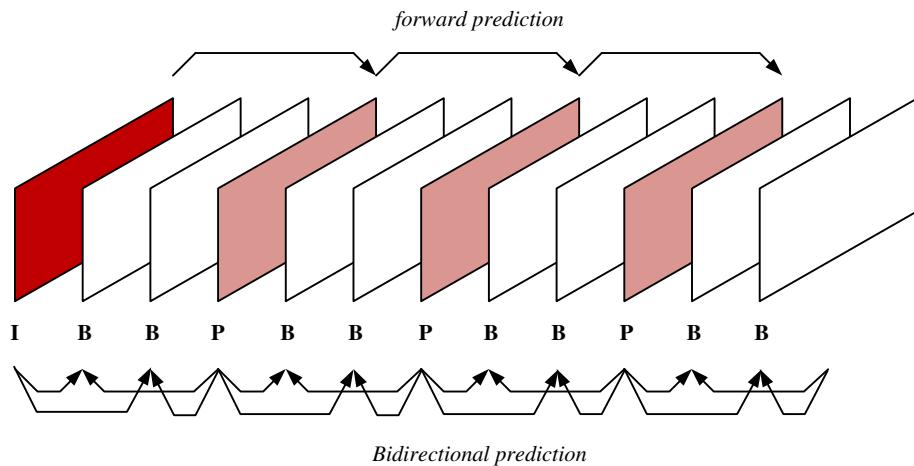
#### 5.01.8.1.1 Prediction modes

Four main classes of prediction are used in video compression:

- **Intra-prediction:** Blocks in the picture are predicted spatially from data adjacent to the current block being coded.
- **Forward prediction:** The **reference** picture occurs temporally **before** the **current** picture.
- **Backward prediction:** The reference picture occurs temporally after the current picture.
- **Bidirectional prediction:** Two (or more) reference pictures (forward and backward) are employed and the candidate predictions are combined in some way to form the final prediction.

**FIGURE 1.6**

Probability distributions for an original frame and FD and DFD frames from the *Football* sequence.

**FIGURE 1.7**

Typical group of pictures structure.

### 5.01.8.1.2 Picture types

Three major types of picture (or frame) are employed in most video codecs:

- **I-pictures:** These are intra-coded (coded without reference to any other pictures).
- **P-pictures:** These are inter-coded with forward (or backward) prediction from another I- or P-picture.
- **B-pictures:** These are inter-coded with prediction from more than one I- and/or P-picture.

序列对象  
组成

Coded pictures are arranged in a sequence known as a **Group of Pictures** (GOP). A typical GOP structure comprising 12 frames is shown in Figure 1.7. A GOP will contain one I-picture and zero or more P- and B- pictures. The 12 frame GOP in Figure 1.7 is sometimes referred to as an IBBPBBPBBPBB structure and it is clear, for reasons of causality, that the encoding order is different to that shown in the figure since the P-pictures must be encoded prior to the preceding B-pictures.

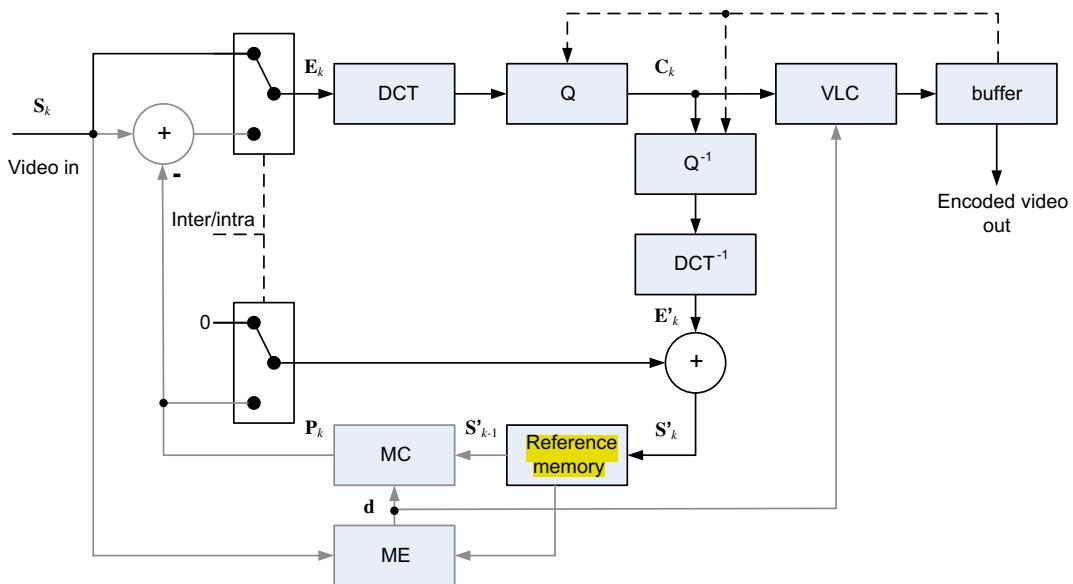
## 5.01.8.2 Operation of the video encoder

### 5.01.8.2.1 Intra-mode encoding

A generic structure of a video encoder is given in Figure 1.8. We first describe the operation of the encoder in intra-mode:

1. The inter/intra switch is placed in the intra position.
2. A forward decorrelating transform is performed on the input frame which is then quantized according to the prevailing rate-distortion criteria:  $\mathbf{C}_k = Q(\text{DCT}(\mathbf{E}_k))$ .
3. The transformed frame is then entropy coded and transmitted to the channel.

编码流程

**FIGURE 1.8**

The block-based motion-compensated video encoder.

4.  $C_k$  is then inverse quantized and inverse DCT'd to produce the same decoded frame pixel values as at the decoder:  $\mathbf{E}'_k = \text{DCT}^{-1}(Q^{-1}(\mathbf{C}_k))$ .
5. The reference memory is finally updated with the reconstructed frame:  $\mathbf{S}'_k = \mathbf{E}'_k$ .

#### **5.01.8.2.2 Inter-mode encoding**

After the first intra-frame is encoded, the following frames in the GOP will be encoded in inter-mode and this is described below:

1. The inter/intra switch is placed in the inter position.
2. Firstly the motion vector for the current frame is estimated:  $\mathbf{d} = \text{ME}(\mathbf{S}_k, \mathbf{S}'_{k-1})$ .
3. Next the motion compensated prediction frame,  $\mathbf{P}_k$ , is formed:  $\mathbf{P}_k = \mathbf{S}'_{k-1}[\mathbf{p} - \mathbf{d}]$ .
4. This is subtracted from the current frame to produce the displaced frame difference (DFD) signal:  $\mathbf{E}_k = \mathbf{S}_k - \mathbf{P}_k$ .
5. A forward decorrelating transform is then performed on the DFD and the result is quantized according to the prevailing rate-distortion criteria:  $\mathbf{C}_k = Q(\text{DCT}(\mathbf{E}_k))$ .
6. The transformed DFD, motion vectors and control parameters are then entropy coded and transmitted to the channel.
7.  $\mathbf{C}_k$  is then inverse quantized and inverse DCT'd to produce the same decoded frame pixel values as at the decoder:  $\mathbf{E}'_k = \text{DCT}^{-1}(Q^{-1}(\mathbf{C}_k))$ .
8. Finally the reference memory is updated with the reconstructed frame:  $\mathbf{S}'_k = \mathbf{E}'_k + \mathbf{P}_k$ .

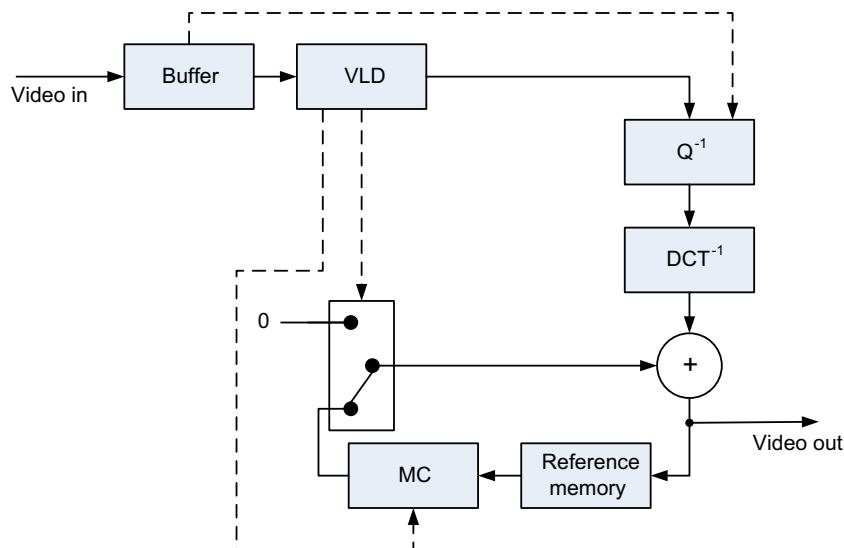
### 5.01.8.3 Operation of the video decoder

The structure of the video decoder is illustrated in [Figure 1.9](#) and described below. By comparing the encoder and decoder architectures, it can be seen that the encoder contains a complete replica of the decoder in its prediction feedback loop. This ensures that (in the absence of channel errors) there is no drift between the encoder and decoder operations. Its operation is as follows, firstly in intra-mode:

1. The inter/intra switch is placed in the intra position.
2. Entropy decoding is then performed on the transmitted control parameters and quantized DFD coefficients.
3. The data  $\mathbf{C}_k$  is inverse quantized and inverse DCT'd to produce the decoded frame pixel values:  $\mathbf{E}'_k = \text{DCT}^{-1}(Q^{-1}(\mathbf{C}_k))$ .
4. Finally the reference memory is updated with the reconstructed frame which is also output to a file or display:  $\mathbf{S}'_k = \mathbf{E}'_k$ .

Similarly in inter-mode:

1. The inter/intra switch is placed in the inter position.
2. Entropy decoding is firstly performed on the control parameters, quantized DFD coefficients, and motion vector.
3.  $\mathbf{C}_k$  is then inverse quantized and inverse DCT'd to produce the decoded DFD pixel values:  $\mathbf{E}'_k = \text{DCT}^{-1}(Q^{-1}(\mathbf{C}_k))$ .



**FIGURE 1.9**

The block-based motion-compensated video decoder.

4. Next the motion compensated prediction frame,  $\mathbf{P}_k$ , is formed:  $\mathbf{P}_k = \mathbf{S}'_{k-1}[\mathbf{p} - \mathbf{d}]$ .
5. Finally the reference memory is updated with the reconstructed frame and this is also output to a file or display:  $\mathbf{S}'_k = \mathbf{E}'_k + \mathbf{P}_k$ .

### 5.01.9 Standardization of video coding systems

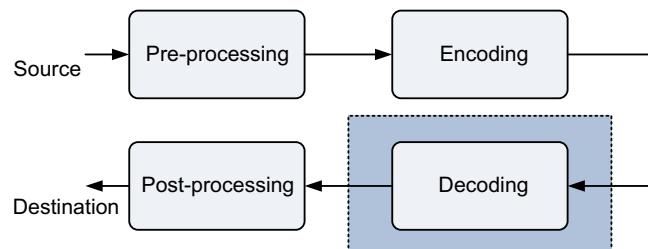
Standardization of image and video formats and compression methods has been instrumental in the success and universal adoption of video technology. An overview of coding standards is provided below and a more detailed description of the primary features of the most recent standard (HEVC) is provided in [Chapter 3](#).

Standards are essential for interoperability, enabling material from different sources to be processed, and transmitted over a wide range of networks or stored on a wide range of devices. This interoperability opens up an enormous market for video equipment, which can exploit the advantages of volume manufacturing, while also providing the widest possible range of services for users. Video coding standards define the bitstream format and decoding process, not (for the most part) the encoding process. This is illustrated in [Figure 1.10](#). A standard-compliant encoder is thus one that produces a compliant bitstream and a standard-compliant decoder is one that can decode a standard-compliant bitstream. The real challenge lies in the bitstream generation, i.e. the encoding, and this is where manufacturers can differentiate their products in terms of coding efficiency, complexity, or other attributes. Finally it is important to note that the fact that an encoder is standard-compliant, provides no guarantee of absolute video quality.

#### 5.01.9.1 A brief history of video encoding standards

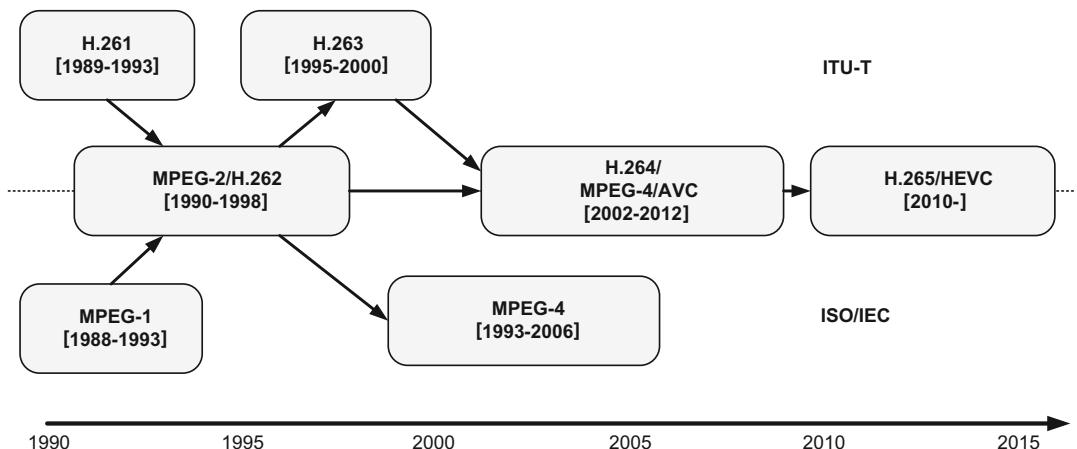
A chronology of video coding standards is represented in [Figure 1.11](#). This shows how the International Standards Organization (ISO) and the International Telecommunications Union (ITU-T) have worked both independently and in collaboration on various standards. In recent years, most ventures have benefited from close collaborative working.

Study Group SG.XV of the CCITT (now ITU-T) produced the first international video coding standard, H.120, in 1984. H.120 addressed videoconferencing applications at 2.048 Mbps and 1.544 Mbps



**FIGURE 1.10**

The scope of standardization.

**FIGURE 1.11**

A chronology of video coding standards from 1990 to the present date.

for 625/50 and 525/60 TV systems respectively. This standard was never a commercial success. H.261 [17] followed this in 1989 with a codec based on a  $p \times 64$  kbps ( $p = 1 \dots 30$ ) targetted at ISDN conferencing applications. This was the first block-based hybrid compression algorithm using a combination of transformation (the Discrete Cosine Transform (DCT)), temporal Differential Pulse Code Modulation (DPCM), and motion compensation. This architecture has stood the test of time as all major video coding standards since have been based on it.

In 1988 the Moving Picture Experts Group (MPEG) was founded, delivering a video coding algorithm targeted at digital storage media at 1.5 Mbs/s in 1992. This was followed in 1994 by MPEG-2 [18], specifically targetted at the emerging digital video broadcasting market. MPEG-2 was instrumental, through its inclusion in all set-top boxes for more than a decade, in truly underpinning the digital broadcasting revolution. A little later in the 1990s ITU-T produced the H.263 standard [19]. This addressed the emerging mobile telephony, internet, and conferencing markets at the time. Although mobile applications were slower than expected to take off, H.263 had a significant impact in conferencing, surveillance, and applications based on the then-new Internet Protocol.

MPEG-4 [20] was a hugely ambitious project that sought to introduce new approaches based on object-based as well as, or instead of, waveform-based methods. It was found to be too complex and only its Advanced Simple Profile (ASP) was used in practice. This formed the basis for the emerging digital camera technology of the time.

Around the same time ITU-T started its work on H.264 and this delivered its standard, in partnership with ISO/IEC, in 2004 [8]. In the same way that MPEG-2 transformed the digital broadcasting landscape, so has H.264/AVC transformed the mobile communications and internet video domains. H.264/AVC is by far the most ubiquitous video coding standard to date. Most recently in 2013, the joint activities of ISO and ITU-T delivered the HEVC standard [10,21], offering bit rate reductions of up to 50% compared with H.264/AVC.

**Table 1.3** Comparison of Video Coding Standards for Entertainment Applications. Average Bit-Rate Savings are Shown for Equal Objective Quality Values Measured Using PSNR

Video standard	Relative bit rate savings (%)		
	H.264/AVC	MPEG-4	MPEG-2
<b>HEVC MP</b>	35.4	63.7	70.8
<b>H.264/AVC HP</b>	X	44.5	55.4
<b>MPEG-4 ASP</b>	X	X	19.7
<b>H.263 HLP</b>	X	X	16.2

Further extensions to video compression standards have been developed that enable the efficient processing of new formats, in particular more immersive formats such as stereoscopic 3-D. The compression challenges and architectures associated with stereoscopic and multiview coding are described in detail in [Chapter 4](#).

### 5.01.9.2 The performance of current standards

An excellent comparison of coding standards is provided by Ohm et al. in [21], where a comprehensive comparison between HEVC and previous standards is reported. We reproduce their results here in [Table 1.3](#), noting that even greater improvements were reported for interactive applications. As a rule of thumb video coding standards have, since the introduction of H.120 in 1984, delivered a halving of bit rate for the equivalent video quality every 10 years. This is evidenced by the results in [Table 1.3](#).

---

## 5.01.10 Conclusions

This chapter has introduced the **requirements** for, and **applications** of video compression. It has examined the **architecture** of a compression system in the context of modern communication systems and has shown that its design is often a **compromise** between coding rate, picture quality, and implementation complexity. The basic operation of a block-based motion compensated video encoding system has been described, explaining its major components, and its operating characteristics. Finally the justification for universal compression standards has been presented, and the performance of recent standards compared in terms of their rate-distortion characteristics.

The remainder of this section will cover many of these topics in more detail and place them in the context of current and future standards, formats, and transmission requirements. [Chapter 2](#) expands on the important area of motion estimation, demonstrating efficient means of achieving temporal decorrelation in the coding process. [Chapter 3](#) provides the reader with an overview of the new HEVC standard and [Chapter 4](#) considers the extensions to compression mechanisms that are required to deal with multiview and stereoscopic content. [Chapters 5](#) and [6](#), provide an insight into future coding possibilities based on an increased awareness of perceptual properties and limitations. [Chapter 7](#) covers the important

area of how we can assess video quality, both for comparing the performance of different codecs and for optimizing coding decisions within the compression process. Finally, [Chapters 8](#) and [9](#) address the topics of content delivery, network adaptation, and error concealment.

---

## Additional resources

1. <http://www.poynton.com/Poynton-video-eng.html>. Lots of information on color conversion, formats, and video preprocessing.
  2. <http://mpeg.chiariglione.org/>. This is the home page of the Moving Picture Experts Group (MPEG), a working group of ISO/IEC with the mission to develop standards for coded representation of digital audio and video and related data. Lots of information on standards with tutorial content.
- 

## Glossary of terms

1080p30	a means of representing the sampling structure of the video format. In this case representing an HD format of 1080 lines with progressive temporal sampling at 30 fps
Discrete cosine transform	a popular decorrelating transform used in image and video coding. Achieves close to optimum performance with a fixed set of basis functions
Entropy coding	the process used to efficiently encode symbols formed from scans of quantized transform coefficients (or motion vectors). A reversible process, usually performed using Huffman or arithmetic coding
Error resilience	the process of making an encoded bitstream more robust to the effects of loss during transmission
Lossless coding	a reversible coding process where the original signal can be exactly reconstructed after compression
Lossy coding	an irreversible coding process where the original signal cannot be exactly reconstructed after compression. Distortions are introduced during coding
Macroblock	the basic coding unit used in many standards. Typically comprising (for 4:2:0) a luma block of 16 by 16 samples and two chroma blocks each of 8 by 8 samples
Motion estimation	the process of temporally predicting the current picture based on the content of other pictures in the sequence
Quantization	the process of approximating the output from the decorrelating transform. A primary mechanism for achieving rate-distortion trade-offs during coding
Streaming	compressed video delivery from a server. Media is constantly received by and presented to an end-user while being delivered
Zettabyte	$10^{23}$ bytes
X:Y:Z	a means of describing the color subsampling method used in a format or by a codec. e.g. 4:2:0 means that the chroma (color difference signals are subsampled by a factor of two horizontally and vertically prior to coding or transmission

---

## References

- [1] D. Bull, *Communicating Pictures*, Academic Press, 2014.
- [2] <<http://www.wireless-world-research.org/fileadmin/sites/default/files/publications/-Outlook/Outlook4.pdf>>.
- [3] Cisco Visual Networking Index: Forecast and Methodology, 2011–2016 (update 2012–2017). <[http://www.cisco.com/en/US/netsol/ns827/networking\\_solutions\\_sub\\_solution.html](http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html)>.
- [4] Network Camera and Video Analytics Market – Global Forecast, Trend & Analysis – Segmentation by Technology, Function, Resolution, Product & Service Type, System Architecture, Verticals, Application and Geography (2012–2017) Report by [marketsandmarkets.com](http://marketsandmarkets.com), Report Code: SE 1238, 2012.
- [5] A. Ortega, K. Ramchandran, Rate-distortion methods for image and video compression, *IEEE Signal Process. Mag.* 15 (6) (1998) 23–50.
- [6] G. Sullivan, T. Wiegand, Rate-distortion optimization for video compression, *IEEE Signal Process. Mag.* 15 (6) (1998) 74–90.
- [7] Recommendation ITU-R BT.2020 (08/2012), Parameter Values for Ultra-High Definition Television Systems for Production and International Programme Exchange, ITU-R, 2012.
- [8] ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), Version 1, 2003; Version 2, 2004; Versions 3, 4, 2005; Versions 5, 6, 2006; Versions 7, 8, 2007; Versions 9, 10, 11, 2009; Versions 12, 13, 2010; Versions 14, 15, 2011; Version 16, 2012.
- [9] ISO/IEC International Standard 10918-1, Information Technology – Digital and Coding of Continuous-Tone Still Images – Requirements and Guidelines, 1992.
- [10] G. Sullivan, J.-R. Ohm, W. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Trans. Circ. Syst. Video Technol.* 22 (12) (2012) 1648–1667.
- [11] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 ISO/IEC 23008-2 and ITU-T Recommendation H.265, High Efficiency Video Coding (HEVC), January 2013.
- [12] N. Ahmed, T. Natarajan, K. Rao, Discrete cosine transform, *IEEE Trans. Comput.* 23 (1974) 90–93.
- [13] D.A. Huffman, A method for the construction of minimum-redundancy codes, *Proc. IRE* 40 (9) (1952) 1098–1101.
- [14] N. Abramson, *Information Theory and Coding*, McGraw-Hill, 1963.
- [15] J. Rissanen, Generalized Kraft inequality and arithmetic coding, *IBM J. Res. Dev.* 20 (1976) 198–203.
- [16] A. Said, Introduction to arithmetic coding – theory and practice, in: K. Sayood (Ed.), *Lossless Compression Handbook*, Academic Press, 2003.
- [17] Int. Telecommun. Union-Telecommun. (ITU-T), Recommendation H.261, Video Codec for Audiovisual Services at  $p \times 64$  kbit/s, Version 1, 1990; Version 2, 1993.
- [18] ITU-T and ISO/IEC JTC 1, Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), Version 1, 1994.
- [19] ITU-T, Video Coding for Low Bitrate Communication, ITU-T Rec. H.263, Version 1, 1995; Version 2, 1998; Version 3, 2000.
- [20] ISO/IEC JTC 1, Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual), Version 1, 1999; Version 2, 2000; Version 3, 2004.
- [21] J.-R. Ohm, G. Sullivan, H. Schwartz, T. Tan, T. Wiegand, Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC), *IEEE Trans. Circ. Syst. Video Technol.* 22 (12) (2012) 1669–1684.

# Motion Estimation—A Video Coding Viewpoint

**Béatrice Pesquet-Popescu, Marco Cagnazzo, and Frédéric Dufaux**

*Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI, 46 rue Barrault, 75634 Paris Cedex 13, France*

---

## Nomenclature

2D	two-dimensional
3D	three-dimensional
<b>AE</b>	Angular Error
AVC	Advanced Video Coding
DCT	Discrete Cosine Transform
DFD	Displaced Frame Difference
<b>DWT</b>	Discrete Wavelet Transform
<b>EE</b>	Error in flow Endpoint
FFT	Fast Fourier transform
<b>FIR</b>	Finite Impulse Response
<b>GMC</b>	Global Motion Compensation
GME	Global Motion Estimation
HEVC	High Efficiency Video Coding
HVS	Human Visual System
<b>HSV</b>	Hue, Saturation, Value
IE	Interpolation Error
<b>IIR</b>	Infinite Impulse Response
LMC	Local Motion Compensation
LME	Local Motion Estimation
<b>LMedS</b>	Least Median of Squares
<b>LMS</b>	Least Mean Square
<b>LTS</b>	Least-Trimmed Squares
MAE	Mean Absolute Error
MB	MacroBlock
MC	Motion Compensation
ME	Motion Estimation
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MVF	Motion Vector Field

NE	Normalized interpolation Error
<b>OBMC</b>	Overlapped Block Motion Compensation
PB	Prediction Block
PSNR	Peak Signal-to-Noise Ratio
RD	Rate-Distortion
RDO	Rate-Distortion Optimization
RMS	Root-Mean-Square
<b>RS</b>	Recursive Search
SAD	Sum of Absolute Differences
SSD	Sum of Squared Differences
<b>SSIM</b>	Structural SIMilarity
<b>SVD</b>	Singular Value Decomposition
TV	Total Variation
<b>VIF</b>	Visual Information Fidelity
VOP	Video Object Plan
<b>ZN-SSD</b>	Zero-mean Normalized SSD

### 5.02.1 Introduction

Digital video is becoming ubiquitous, thanks to tremendous technological progress over recent decades, with widespread applications in information technology, telecommunications, consumer electronics, and entertainment.

**地位** In video sequences, motion is a **key source** of information. Motion arises due to **moving objects** in the 3D scene, as well as **camera motion**. Apparent motion, also known as **optical flow**, captures the resulting spatio-temporal variations of pixel intensities in successive images of a sequence. The purpose of motion estimation techniques is to **recover this information by analyzing the image content**. Efficient and accurate motion estimation is an essential component in the domains of image sequence analysis, computer vision, and video communications.

**来源2** In the context of image sequence analysis and computer vision, the objective of motion estimation algorithms is to precisely and faithfully model the motion in the scene. This information is fundamental for video understanding and object tracking. Relevant applications include video surveillance, robotics, autonomous vehicle navigation, human motion analysis, quality control in manufacturing, video search and retrieval, and video restoration. Accurate motion is also important in some video processing tasks such as frame rate conversion or de-interlacing of television formats.

**目的** As far as video coding is concerned, compression is achieved by **exploiting data redundancies in both the spatial and temporal dimensions**. **Spatial redundancy reduction** is largely achieved by **transform coding**, e.g., using the Discrete Cosine Transform (DCT) or the Discrete Wavelet Transform (DWT), which effectively compacts the signal energy into a few significant coefficients. In turn, **temporal redundancies** are reduced by means of **predictive coding**. Observing that **temporal correlation** is maximized along **motion trajectories**, motion compensated prediction is used for this purpose. In this context, the main objective of motion estimation is **no longer to find the “true” motion in the scene, but rather to maximize**

**压缩实现**

**S, T,  
transform  
predictive**

**分布，主要  
来源，组成**

**目的**

实现达成  
提供

compression efficiency. In other words, motion vectors should provide a precise prediction of the signal. Moreover, the motion information should enable a compact representation, as it has to be transmitted as overhead in the compressed code stream. Efficient motion estimation is key to achieve high compression in video coding applications such as TV broadcasting, Internet video streaming, digital cinema, DVD, Blu-ray Disc, and video-conferencing.

二元性

We should also note a duality between motion estimation and segmentation operations. More specifically, in order to correctly estimate motion, regions of homogeneous motion need to be known. Conversely, for accurate segmentation of these regions, it is necessary to previously perform motion estimation. This problem can be tackled by joint motion estimation and segmentation techniques involving iterative algorithms. However, in this chapter, we will exclusively focus on the motion estimation aspects.

As a final remark, perception of motion by the Human Visual System (HVS) is also an important topic. Measurement and interpretation of visual motion is discussed in [1,2]. Better understanding of human perception could help to improve current motion estimation techniques, or lead to new approaches.

Based on the above discussion, motion estimation is clearly a vast and complex topic. The purpose of this chapter is to give a broad overview of motion estimation techniques with a special emphasis on video compression requirements.

As a complement to this chapter, readers may refer to earlier surveys, including [3–6].

## 5.02.2 Motion representation and models

### 5.02.2.1 2D motion vector field and optical flow

While we all have an intuitive understanding of the concept of motion, this notion deserves to be clarified in the case of digital video processing.

认识，投影

Motion is unambiguously defined in the physical three-dimensional (3D) world. However, when capturing an image, a two-dimensional (2D) projection of the 3D scene is performed. Straightforwardly, the motion arising in an image sequence is also the direct product of the projection of an object's displacement in the 3D scene.

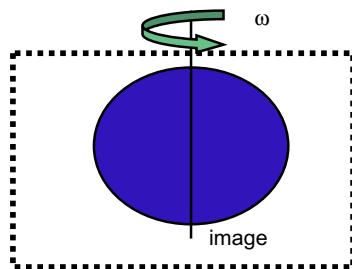
In particular, we can make a distinction between two different concepts: the 2D motion vector field and the optical flow. More specifically, the 2D motion vector field is defined as the projection of the 3D object's motion onto the 2D image plane [7]. In contrast, the optical flow is defined as apparent motion of the brightness pattern [7,8]. In other words, the optical flow captures the spatio-temporal variation of pixel intensities.

关系，部分伴生

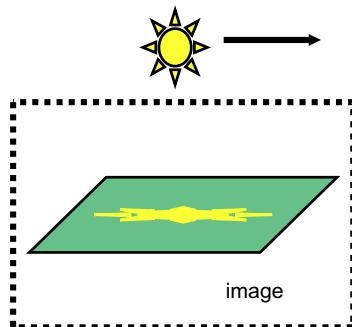
The 2D motion vector field and the optical flow often coincide, although it is not mandatory. To better understand the difference, let us consider two simple examples. Figure 2.1 shows a three-dimensional uniform sphere in pure rotation and under constant illumination. In this case, the 2D motion vector field is non-zero. However, this motion leads to a constant brightness pattern, and therefore the optical flow is zero. Conversely, Figure 2.2 depicts a static reflecting surface, illuminated by a moving light source. In this case, the 2D motion vector field is zero, whereas the optical flow is non-zero.

目的

In this chapter, we specifically consider motion estimation techniques for video coding. The objective is to estimate the displacement of pixels between successive images in order to predict the pixel intensities, in other words the optical flow is estimated. However, for the sake of simplicity, we will

**FIGURE 2.1**

Three-dimensional uniform sphere in rotation under constant illumination.

**FIGURE 2.2**

Static reflecting surface illuminated by a moving light source.

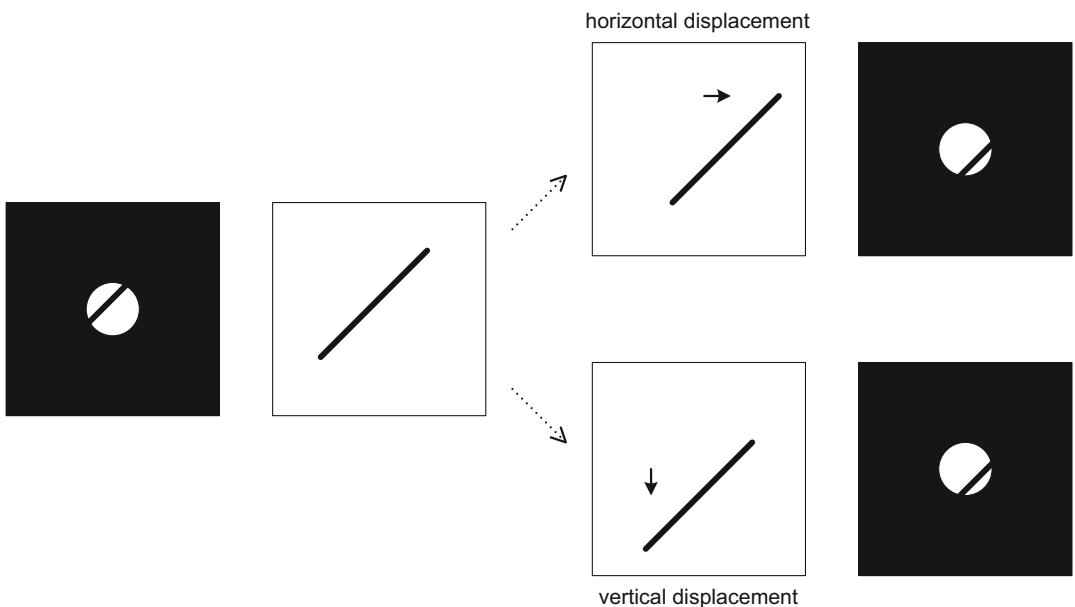
interchangeably use the terms 2D motion vector field and optical flow throughout the chapter as this is commonly the case in the video processing community.

### 5.02.2.2 The aperture problem

**问题** The aperture problem occurs when observing a moving structure through a small aperture. More specifically, under this condition, different physical motions appear indistinguishable. This phenomenon is illustrated in Figure 2.3. A bar is moving horizontally (top) or vertically (bottom). When seen through a small aperture, which only shows a part of the whole object, both the horizontal and vertical displacements produce the same appearance and therefore cannot be distinguished (right-hand side).

**原因** The same aperture problem happens with motion-sensitive neurons in the visual primary cortex [1]. These neurons, with a finite receptive field, always react to a moving contour that passes through their receptive field, as long as it is consistent with the neuron's preferred direction and regardless of the true motion orientation.

### 生理现象

**FIGURE 2.3**

Aperture problem: when observing a moving structure through a small aperture, different physical motions appear indistinguishable: (top-right) bar with horizontal displacement seen through aperture, (bottom-right) bar with vertical displacement seen through aperture.

### 5.02.2.3 Motion representation

In order to define a motion representation, two issues have to be considered. Firstly, a representative model of the motion needs to be specified. Secondly, the region of support to which the model applies has to be identified. We discuss these two subjects in more detail hereafter.

#### 5.02.2.3.1 Brightness constancy motion model and the optical flow equation

表示：  
连续，离散

An **image sequence** can be considered as a three-dimensional continuous spatio-temporal field:  $f(x, y, t)$ , where  $(x, y)$  are the spatial coordinates and  $t$  is the time index. However, in practice we only dispose of a discrete version of this function:

$$f_{n,m,k} = f(nL_1, mL_2, kT), \quad (2.1)$$

where  $L_1$  and  $L_2$  are the sampling steps in the spatial domain,  $T$  is the temporal sampling step, and  $n$ ,  $m$  and  $k$  are integers.

假设

In what follows, we shall make the following **assumptions**:

- A pixel intensity remains unchanged along a motion trajectory. This assumption is known as the **brightness constancy constraint**. In other words, the variations in time of the pixel intensities are due

to the displacements of different objects present in the scene. The brightness constancy constraint implies that the illumination is uniform and the scene is Lambertian.

- The motion appears **locally** as a translation (which is the simplest, though effective, motion model).

We then have the following brightness constancy motion model:

$$f(x, y, t + T) = f(x - \Delta x, y - \Delta y, t), \quad (2.2)$$

where  $\Delta x$  and  $\Delta y$  are, respectively, the horizontal and vertical components of the displacement. Actually, these components depend on the spatial position, the temporal index  $t$ , and the temporal sampling step  $T$ , but for the sake of simplicity, we will not make this dependence appear explicitly in the following.

We can now derive the ***motion constraint*** or the ***optical flow equation***. A first-order **Taylor expansion** of the motion model defined in Eq. (2.2) leads to:

$$\begin{aligned} f(x, y, t + T) &= f(x, y, t) - \Delta x \frac{\partial f}{\partial x}(x, y, t) - \Delta y \frac{\partial f}{\partial y}(x, y, t) \\ &\quad + o(|\Delta x| + |\Delta y|). \end{aligned} \quad (2.3)$$

Let us define the displacement vector  $D$  as

$$D = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}. \quad (2.4)$$

We remark that the vector containing the partial derivatives of the illumination field  $f$  with respect to the coordinates  $x$  and  $y$  is nothing else than the spatial gradient of the image,

$$\nabla f(x, y, t) = \begin{pmatrix} \frac{\partial f}{\partial x}(x, y, t) \\ \frac{\partial f}{\partial y}(x, y, t) \end{pmatrix}. \quad (2.5)$$

Then we can write the same equation in a vector form, as follows:

$$f(x, y, t + T) = f(x, y, t) - D^T \nabla f(x, y, t) + o(\|D\|), \quad (2.6)$$

where  $D(x, y)^T \nabla f(x, y, t)$  is the inner product of the two vectors.

Now, we define the ***velocity vector field*** in the sequence, also called ***optical flow***, as:

$$V = \begin{pmatrix} u \\ v \end{pmatrix} = \lim_{T \rightarrow 0} \frac{D}{T}. \quad (2.7)$$

With this definition, the original equation can be re-written as:

$$V^T \nabla f(x, y, t) + \frac{\partial f}{\partial t}(x, y, t) = 0, \quad (2.8)$$

and if we develop it, we can write:

$$u(x, y) \frac{\partial f}{\partial x}(x, y, t) + v(x, y) \frac{\partial f}{\partial y}(x, y, t) + \frac{\partial f}{\partial t}(x, y, t) = 0. \quad (2.9)$$

This equation is essential in all motion field estimation. It is called the *motion constraint* or the *optical flow equation*.

We can make the two following observations:

- The determination of the real displacement in the sequence from the motion constraint is **under-determined**, since we dispose of only **one** equation and we have **two** unknown variables (the optical flow components  $u$  and  $v$ ). This problem, known as the **aperture problem** and already introduced in [Section 5.02.2.2](#), can be easily understood from the vector form of the equation. In this case, we have an inner product between  $V$  and the gradient of the image. Clearly, only the component of the velocity vector **parallel to** the gradient can be determined. The other component is cancelled in the inner product.
- In order to solve this ambiguity, one generally has to **add an homogeneity constraint** for the displacement. Some methods to do this will be discussed in [Section 5.02.3](#).

欠定  
原因  
何时可定

### 5.02.2.3.2 Parametric motion models

Another approach is to model motion fields by a set of parameters. Such a model is efficient in representing the motion of a region whose pixels have a **coherent motion**. Moreover, parametric models result in very compact descriptors, as only a small set of parameters have to be transmitted.

A simple model can be formed by considering, in the image plane, a **polynomial approximation** of the displacement [6]. More formally, this polynomial model can be expressed as

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \sum_{i,j} \begin{pmatrix} a_{i,j} \\ b_{i,j} \end{pmatrix} x^i y^j, \quad (2.10)$$

where  $d_x$  and  $d_y$  are the two components of the motion vector,  $a_{i,j}$  and  $b_{i,j}$  represent the parameters of the model, and  $(x, y)$  are the pixel coordinates.

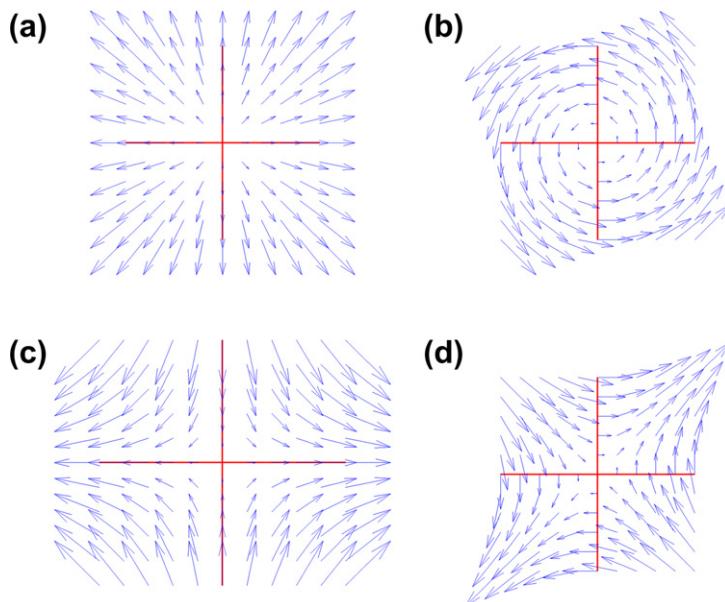
For example, a first-order approximation leads to a description of the form

$$\begin{aligned} d_x &= t_x + (k + h_1)x + (h_2 - a)y, \\ d_y &= t_y + (h_2 + a)x + (k - h_1)y, \end{aligned} \quad (2.11)$$

where  $t_x$ ,  $t_y$ ,  $a$ ,  $k$ ,  $h_1$ , and  $h_2$  are the motion model parameters. In this example,  $t_x$  and  $t_y$  represent the translation parameters. The parameters  $a$ ,  $x_0$ ,  $y_0$ ,  $k$ ,  $h_1$ , and  $h_2$  are involved in the description of more complex displacements. For instance, as illustrated in [Figure 2.4](#), a **small rotation** can be described uniquely using the parameter  $a$ , a **divergent** motion vector field will be described by  $k$ , and a **hyperbolic** motion will be represented using  $h_1$  and  $h_2$ .

几种含义

Similar interpretations can be obtained by representing the three-dimensional motions of a rigid body in the space and projecting, under some hypotheses, these motions on the two-dimensional image plane. These hypotheses concern a small **vision angle**, small **displacements** orthogonal to the projection plane, and small **rotations** around the axes in the image plane [6].

**FIGURE 2.4**

Parametric models for motion vector fields : (a) divergence ( $k = 0.5$ ); (b) rotation ( $a = 0.5$ ); (c) hyperbolic field ( $h_1 = 0.5$ ); (d) hyperbolic field ( $h_2 = 0.5$ ).

From this standpoint, polynomial models can be derived:

**几种运动模式**

- **Translational** motion model:

$$\begin{aligned} d_x &= a_1, \\ d_y &= b_1. \end{aligned} \quad (2.12)$$

In this case, a very simple zero-order polynomial form is obtained. It can be derived from a rigid translational 3D motion under orthographic projection. However, this model fails to take into account **zoom**, **rotation**, **pan**, and **tilt** of the camera. Nevertheless, thanks to its simplicity, it is widely used in **block matching** motion estimation techniques (see [Section 5.02.6](#)). Straightforwardly, any 2D shape is preserved after motion compensation using such a model.

- **Affine** motion model:

$$\begin{aligned} d_x &= a_1 + a_2x + a_3y, \\ d_y &= b_1 + b_2x + b_3y. \end{aligned} \quad (2.13)$$

A first-order polynomial form, the affine motion model can be derived from the 3D affine motion of a planar surface under orthographic projection. With the affine model, motion compensation conserves **parallel lines**.

**相机运动情况**

**适用**

- Perspective (or projective) motion model:

$$\begin{aligned} d_x &= \frac{a_1 + a_2x + a_3y}{1 + a_4x + b_4y}, \\ d_y &= \frac{b_1 + b_2x + b_3y}{1 + a_4x + b_4y}. \end{aligned} \quad (2.14)$$

Considering the 3D affine motion of a planar surface under perspective projection leads to the perspective (or projective) motion model with 8 parameters. Clearly, the affine model is a special case of the perspective one, with  $a_4 = b_4 = 0$ . With the perspective model, lines remain lines after motion compensation. One drawback of this model is the difficulty to accurately estimate the parameters in the denominator term.

- Quadratic motion model:

$$\begin{aligned} d_x &= a_1 + a_2x + a_3y + a_4xy + a_5x^2 + a_6y^2, \\ d_y &= b_1 + b_2x + b_3y + b_4xy + b_5x^2 + b_6y^2. \end{aligned} \quad (2.15)$$

This second-order polynomial form can be derived from a 3D affine motion of a parabolic surface under orthographic projection. Again, the affine model is a special case, with  $a_4 = a_5 = a_6 = b_4 = b_5 = b_6 = 0$ . The perspective model is also included as a Taylor approximation. Note that the quadratic model does no longer preserve lines after motion compensation.

- Bilinear motion model:

$$\begin{aligned} d_x &= a_1 + a_2x + a_3y + a_4xy, \\ d_y &= b_1 + b_2x + b_3y + b_4xy. \end{aligned} \quad (2.16)$$

The bilinear model is obtained from the quadratic model by discarding square terms. However, it is not related to any physical 3D motion.

## 分析

Clearly, the more complex a motion model, the better its ability to precisely represent more complex motions. However, the number of parameters is also higher. Therefore, it makes the estimation process more difficult and complex, and possibly prone to errors. Moreover, a complex motion model does not allow for a compact representation of the motion information.

### 5.02.2.3.3 Region of support

**适用区域** The region of support is the set of image pixels to which a motion model applies. We can distinguish the following four cases:

- **Pixel-based:** A motion vector is assigned to each pixel of the image, resulting in a dense motion field. It has the advantage to provide a precise description of the motion. However, from a video coding viewpoint, it entails a costly representation resulting in a large overhead for motion information.
- **Region-based:** The motion model is applied to a region of the image which is characterized by a coherent motion. In this case, moving objects in the scene have to be identified. In [9], a method is presented for segmenting video scenes hierarchically into differently moving objects. A system for representing moving images with sets of overlapping layers using motion analysis is proposed

找到运动物体  
分割

要求  
in [10]. Segmentation-based motion estimation and spatio-temporal segmentation are addressed in [11]. In [12], a Bayesian framework is presented that combines motion estimation and segmentation. In the context of video coding, such a representation requires to transmit the shape of the region, which entails a bit rate overhead.

- 优势  
  - **Block-based:** As a special case of the region-based support, a very frequent choice is to simply partition the image into blocks. If the block size is sufficiently small, then the assumption that the block is moving in a coherent way is likely to be valid. Another advantage of a block partitioning is that it does not require additional information to represent the shape of the region.
  - **Global:** The region of support simply encompasses the whole image. This case is especially suited to efficiently estimate camera motion. Camera motions, such as dolly, track, boom, pan, tilt, or roll, are all essential cinematic techniques.

关系  
The choices of the region of support and the motion model are closely intertwined. When using a complex parametric motion model, which can handle complex motions, a larger region of support can effectively be used. Conversely, a simple model is often sufficient in conjunction with a small region of support.

#### 5.02.2.3.4 Considerations from a video coding viewpoint

矛盾  
We can clearly observe that, for video coding applications, selection of an optimal motion representation brings up contradictory requirements. On the one hand, the representation has to accurately characterize the motion information, even in the case of complex scenes with multiple moving objects. On the other hand, the representation needs to be compact for further efficient coding, transmission, or storage.

主次  
From a video coding viewpoint, the first two criteria should lead, foremost, to a good prediction (in other words, small prediction error), and simultaneously, to a low overhead information (therefore, easy-to-encode motion vector fields). This tradeoff shows that the ultimate goal is not to obtain the real motion, even though a motion field close to the true motion in the scene avoids artificial discontinuities and reduces the transmission cost of the motion information, but to globally optimize the video coding scheme in a rate-distortion sense.

最终目的，  
不是一方面  
是权衡  
常用  
In practice, in video coding, most schemes are based on a translational motion model combined with a block-based partitioning, as a good tradeoff between motion accuracy and the overhead to represent motion information. However recently it has been shown that region-based motion description can be competitive in a rate-distortion sense [13].

---

### 5.02.3 Optical flow approaches

基于  
In this section, we present optical flow estimation approaches, which have been mainly developed for image sequence analysis applications and computer vision [7]. The methods entering this category are mainly based on gradient techniques, with the common objective to solve the motion constraint or optical flow equation, as defined in Section 5.02.2.3.1. More precisely, we present two classical optical flow algorithms: Horn-Schunck [8] and Lucas-Kanade [14].

### 5.02.3.1 Horn-Schunck

We now present one of the methods using this differential approach, based on a global optimization.

The method we describe here uses a minimization of a cost function, which combines the optimization of the optical flow constraint, as defined in Eq. (2.9), with a constraint on the smoothness of the motion vector field. This method is referred to as the *Horn-Schunck* algorithm [8].

The cost function  $J_{HS}(V)$  is defined as follows:

$$J_{HS}(V) = \iint_{\mathfrak{M}} \left[ u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} + \frac{\partial f}{\partial t} \right]^2 dx dy + \lambda \iint_{\mathfrak{M}} [\|\nabla u\|^2 + \|\nabla v\|^2] dx dy, \quad (2.17)$$

where  $\mathfrak{M}$  is the support on which the optimization is performed, namely it can be a region or the entire image, and  $\lambda$  is a positive constant.

The first term in the above expression is a mean square error on the motion constraint, while the second one is a regularization term: it ensures that the gradient of the motion vector field takes small values (“smoothness” of the solution). In this criterion,  $\lambda$  is the regularization constant, which enables a tradeoff between the influence of the regularization term and the minimization of the motion constraint. Remark that the integrals are performed over an arbitrary region where the motion is homogeneous, meaning that this technique can be adapted to region-based motion estimation or to a joint segmentation-motion estimation solution.

After some mathematical developments involving the minimization of the cost function, we arrive at the following solution for the two components of the motion vector field:

$$\lambda \nabla^2 u = \frac{\partial f}{\partial x} \left[ u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} + \frac{\partial f}{\partial t} \right], \quad (2.18)$$

$$\lambda \nabla^2 v = \frac{\partial f}{\partial y} \left[ u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} + \frac{\partial f}{\partial t} \right], \quad (2.19)$$

where  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  is the Laplacian operator. The following observations can be made concerning this method:

- This global optimization is quite complex, involving the resolution of a system of partial differential equations.
- The choice of  $\lambda$  is critical. On the one hand, it leads to a different smoothness of the field. On the other hand, it influences the numerical stability of the system.

复杂度  
参数

### 5.02.3.2 Lucas-Kanade

The *Lucas-Kanade* method is another classical approach for optical flow estimation [14]. In this method, the assumption is made that the optical flow is approximately constant in a small neighborhood, instead of adding a smoothness constraint as in the Horn-Schunck algorithm.

More precisely, the motion vector  $V = (u, v)$  is assumed to be **constant** in a local region  $\mathfrak{N}$  around the pixel being processed. In other words, the optical flow equation expressed at all pixel locations  $(x_i, y_i) \in \mathfrak{N}, i = 1, \dots, n$ , leads to a set of  $n$  equations:

$$\begin{aligned} u \frac{\partial f}{\partial x}(x_1, y_1) + v \frac{\partial f}{\partial y}(x_1, y_1) + \frac{\partial f}{\partial t}(x_1, y_1) &= 0, \\ \vdots \\ u \frac{\partial f}{\partial x}(x_n, y_n) + v \frac{\partial f}{\partial y}(x_n, y_n) + \frac{\partial f}{\partial t}(x_n, y_n) &= 0. \end{aligned} \quad (2.20)$$

In the Lucas-Kanade method, the optical flow is then obtained by **Least Squares** minimization of the following cost function  $J_{LK}(V)$

$$J_{LK}(V) = \sum_{i=1 \dots n} w_i \left( u \frac{\partial f}{\partial x}(x_i, y_i) + v \frac{\partial f}{\partial y}(x_i, y_i) + \frac{\partial f}{\partial t}(x_i, y_i) \right)^2, \quad (2.21)$$

where the **weighting factors**  $w_i$  have been introduced to give larger weights to pixel locations  $(x_i, y_i)$  which are near the **region center**.

其他用到的地方

Note that this is also the assumption made in block matching (or region matching) algorithms.

### 5.02.3.3 Discussion

结果-形式  
缺点

**Optical flow** approaches result in a dense motion vector field (one vector per pixel), which is qualitatively interesting for motion analysis applications. However, they also have several **weaknesses**:

- The derivation of the optical flow equation is based on a first-order Taylor series expansion. This only holds under the hypothesis that the **motion between two frames is small**.
- The equations are written for continuous-time and continuous-space variables and require to estimate the image gradient. For solving them, we need to **discretize** these variables. However, this sampling process introduces errors in the solution. In particular, gradient computation is sensitive to noise and is therefore subject to errors.
- The **smoothness constraint** (Horn-Schunck) or the **local uniformity constraint** (Lucas-Kanade) results in a poor accuracy along moving object boundaries.

In order to address the above shortcomings, numerous advances have been proposed resulting in improved performance.

改变损失  
函数

Instead of the above formulation based on an **L2 norm**, the **L1 norm** is also a frequent choice [15], both for the optical flow equation and for the additional constraint (e.g., smoothness). Such cases are referred to as **Total Variation (TV) methods**.

A model to handle changes in **illumination** and **blur** is proposed in [16]. It also includes spatial weighting of the smoothness constraint. Anisotropic smoothness weighting is considered in [17]. The method also applies different weighting to color channels in the HSV color space.

A novel extended coarse-to-fine refinement framework is introduced in [18]. The reliance on the initial flow estimates propagated from a coarser level is reduced. Hence, motion details are better preserved at each scale. An adaptation of the objective function to handle outliers and a new optimization procedure are also proposed.

For a more detailed and in-depth **tutorial** on optical flow techniques, the reader is referred to [4]. An extensive performance **comparison** of several algorithms is given in [19]. More recently, a new set of **benchmarks** and **evaluation** methods for optical flow techniques have been introduced in [20]. A taxonomy of optical flow algorithms is also presented, along with a discussion of recent works. At the time of this writing, the method by Xu et al. [18] is one of the best performing techniques reported in the on-line optical flow evaluation database at <http://vision.middlebury.edu/flow/>.

From a video coding perspective, the fact that a dense motion field is obtained is not necessarily a positive point. Indeed, this field has to be encoded, which may result in a high bit rate overhead.

## 5.02.4 Pel-recursive approaches

Pixel- or pel-recursive approaches were among the earliest methods used to estimate motion with the objective of video coding. Essentially, these techniques **recursively estimate the displacement** which minimizes the Displaced Frame Difference (**DFD**) defined as

$$\text{DFD} = f(n, m, k) - f(n - \Delta x, m - \Delta y, k - 1). \quad (2.22)$$

Note that here we consider the discrete spatial and temporal coordinates. The image is typically scanned in a raster order, performing the recursion on a pel-by-pel basis.

### 5.02.4.1 Netravali-Robbins

The first pel-recursive motion estimation algorithm was proposed by Netravali and Robbins [21]. This method **aims** at **minimizing the square of the DFD** using a **steepest descent** technique. Considering the current pixel position  $(n, m)$  and the motion vector  $D_{n,m}^{(i)}$  at iteration  $i$ , the recursion is defined as

$$D_{n,m}^{(i+1)} = D_{n,m}^{(i)} - \frac{\varepsilon}{2} \nabla_D \text{DFD}^2(n, m, \Delta x, \Delta y), \quad (2.23)$$

where  $\nabla_D$  denotes the gradient with respect to  $D$ , and  $\varepsilon > 0$  is a constant gain. Developing the second term, we straightforwardly obtain

$$\begin{aligned} \nabla_D \text{DFD}^2(n, m, \Delta x, \Delta y) &= 2\text{DFD}(n, m, \Delta x, \Delta y) \\ &\times \nabla f(n - \Delta x, m - \Delta y, t - T). \end{aligned} \quad (2.24)$$

Therefore, by substitution, the motion vector update can be re-written as

$$D_{n,m}^{(i+1)} = D_{n,m}^{(i)} - \varepsilon \text{DFD}(n, m, \Delta x, \Delta y) \times \nabla f(n - \Delta x, m - \Delta y, t - T). \quad (2.25)$$

The iteration from  $i$  to  $i + 1$  can be carried out at one given pixel location, or from one location to the next one.

In order to make a more robust estimation of the DFD, it can be computed in a neighborhood around the pixel  $(n, m)$ . Furthermore, bilinear interpolation is used when the displacement is not an integer number of pixels.

### 5.02.4.2 Cafforio-Rocca

In this section, we describe an improved pel-recursive algorithm, known as the *Cafforio-Rocca algorithm* [22].

At each pixel location, in a raster scan order, the following operations are performed:

**(1) Initialization** of the motion vector

As an initial estimate of the motion vector, the vector obtained at the previous position in the scanning order is chosen. Namely, if  $(n, m)$  is the current position, and  $D_{n,m}$  the motion vector at this position, then the initial estimate will be:

$$D_{n,m}^{(0)} = D_{n-1,m}, \quad (2.26)$$

where  $D_{n-1,m}$  is the motion vector estimated at the previous point in the scanning order.

Alternatives can be considered to compute this initial estimate. For example, one could consider an average of several of the previously estimated motion vectors or any other function of these estimations. Note, however, that the causality induced by the scanning order has to be respected, in order to be able to perform the same operations at the decoder.

**(2) Reliability test** of the initial estimate

After this initialization, the next step is to check that the value of the a priori estimation is correct. This test is necessary in order to take into account the following sources of errors:

- strong variations of the motion vectors from one position to the next, related to objects with different motions;
- the divergence of the motion estimation algorithm itself.

For this purpose, let us consider the criterion built on the following expressions:

$$R(n, m) = \left| f(n, m, k) - f(n - \Delta x^{(0)}, m - \Delta y^{(0)}, k - 1) \right| - |f(n, m, k) - f(n, m, k - 1)|, \quad (2.27)$$

with

$$D_{n,m}^{(0)} = \begin{pmatrix} \Delta x^{(0)} \\ \Delta y^{(0)} \end{pmatrix}. \quad (2.28)$$

In other words, the reliability criterion  $R$  is made of two terms. The first is the estimation error resulting from the a priori initial prediction and the second is the simple inter-image difference. We also define a threshold value  $s > 0$ . If  $(R > s)$ , the motion estimation is re-initialized to 0,

otherwise the a priori estimation at pixel  $(n, m)$ , as defined in step 1, is kept. More specifically, the criterion is:

$$\begin{aligned} \text{if } (R(n, m) > s) \quad D_{n,m}^{(1)} &= 0, \\ \text{otherwise} \quad D_{n,m}^{(1)} &= D_{n,m}^{(0)}. \end{aligned} \quad (2.29)$$

### (3) Refinement of the estimation

This is the most important step of the algorithm, updating the motion vector. The actual value of the motion vector is computed as the initial value (validated or re-initialized by the reliability test) plus an update vector, as follows:

$$D_{n,m} = D_{n,m}^{(1)} + \delta D_{n,m}. \quad (2.30)$$

The update vector is chosen so as to minimize a cost function as a tradeoff between the prediction error and the amplitude of the update vector. Note that it is similar to the one used in Horn-Schunck (see [Section 5.02.3.1](#)). More precisely, the cost function is written as:

$$\begin{aligned} J(\delta D) = & \left( f(n, m, k) - f(n - \Delta x^{(1)} - \delta x, m - \Delta y^{(1)} - \delta y, k - 1) \right)^2 \\ & + \lambda \|\delta D\|, \end{aligned} \quad (2.31)$$

with the previous motion estimate

$$D_{n,m}^{(1)} = \begin{pmatrix} \Delta x^{(1)} \\ \Delta y^{(1)} \end{pmatrix}, \quad (2.32)$$

the update vector

$$\delta D_{n,m} = \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}, \quad (2.33)$$

and  $\lambda$  is a positive regularization constant.

泰勒展开

With a first-order Taylor expansion around the previous estimation, we can write the previous image compensated with the updated vector:

$$\begin{aligned} f(n - \Delta x^{(1)} - \delta x, m - \Delta y^{(1)} - \delta y, k - 1) = \\ f(n - \Delta x^{(1)}, m - \Delta y^{(1)}, k - 1) - \delta D^T \varphi_{n,m} + O \|\delta D\|, \end{aligned} \quad (2.34)$$

where

$$\varphi_{n,m} = \nabla f(n - \Delta x^{(1)}, m - \Delta y^{(1)}, k - 1), \quad (2.35)$$

is the gradient of the previous image, compensated with the initial motion vector. Thus we can re-write the cost function as

$$J(\delta D) = \left[ \varepsilon_{n,m} + \delta D^T \varphi_{n,m} \right]^2 + \lambda \|\delta D\|, \quad (2.36)$$

where we introduced

$$\begin{aligned} \varepsilon_{n,m} = & f(n - \Delta x^{(1)} - \delta x, m - \Delta y^{(1)} - \delta y, k - 1) \\ & - f(n - \Delta x^{(1)}, m - \Delta y^{(1)}, k - 1), \end{aligned} \quad (2.37)$$

which denotes the prediction error using the initial motion vector in order to simplify the notation. The minimum of the cost function with respect to the update vector is obtained by cancelling its derivative. This leads to the following value of the **update vector**:

$$\delta D_{n,m} = -\frac{\varphi_{n,m} \varepsilon_{n,m}}{\lambda + \|\varphi_{n,m}\|^2}. \quad (2.38)$$

Straightforwardly, the last step of the algorithm consists in updating the motion vector with this quantity:

$$D_{n,m} = D_{n,m}^{(1)} - \frac{\varphi_{n,m} \varepsilon_{n,m}}{\lambda + \|\varphi_{n,m}\|^2}. \quad (2.39)$$

The choice of the regularization parameter  $\lambda$  and of the threshold  $s$  is very important and can lead to the divergence of the algorithm when they are not well managed.

#### 5.02.4.3 Discussion

The following remarks can be made regarding pel-recursive approaches such as Netravali-Robbins or Cafforio-Rocca. Firstly, these pel-recursive techniques are relatively **easy to implement**. **Convergence** can be slow and the motion estimation is not always of very good quality. This is especially the case when the **displacement** is important or when there are large **motion discontinuities** at object borders. This **limitation** is mostly due to the recursive nature of the algorithm with a **causality constraint**.

In order to improve convergence and performance, other pel-recursive motion estimation techniques have been proposed. Better adaptation to local statistics is obtained in [23]. A Wiener-based displacement estimation algorithm is introduced in [24]. A multiple frames model-based approach is proposed in [25]. Finally, a multiple mask regularization technique is introduced in [26].

There are some **advantages** of the pel-recursive methods over other existing approaches. Provided that the recursion has a **quick enough convergence** (i.e., it can handle the motion discontinuities), the pel-recursive algorithms may overcome the **problem of multiple moving objects**. Moreover, as the update vector calculation is based only on previously transmitted data (causality), the decoder can estimate the same motion vector field as the encoder. **No overhead information** is thus required for transmitting the motion vector field, which is of course a big advantage of these methods in video coding applications. The **counterpart** is that the decoder has to perform the same operations as the encoder in order to find the motion vectors, which will lead to increased **computational complexity**.

---

#### 5.02.5 Transform-domain approaches

The **basic idea** of transform-domain approaches is to estimate the motion vectors from a measure performed in the transform domain, for instance with a Fourier transform, a Discrete Cosine Transform (DCT), or a Discrete Wavelet Transform (DWT). For this purpose, the effect of the 2D motion on the characteristics of the transform has to be studied.

### 5.02.5.1 Motion estimation in the Fourier or DCT domain

Let us first consider the case of motion estimation in the Fourier domain [27, 28]. In this case, a **translation** in the spatial domain corresponds to a **phase-change** of the Fourier coefficients. Instead of minimizing a dissimilarity, for instance using Mean Square Error (MSE) or Mean Absolute Error (MAE) as in block matching techniques (see [Section 5.02.6](#)), a **phase correlation** between blocks is usually performed. Therefore, these methods are sometimes referred to as **phase-correlation** methods.

A common **drawback** of such methods is that it is difficult to characterize **complex** motions in the transform domain. As a consequence, a **simple** motion model has to be adopted, which may adversely affect the **precision** of the motion estimation process. Nevertheless, correlation-based methods have been successfully applied for the estimation of **global motion** and have been widely used in video standards conversion.

Similarly, the same ideas can be applied in the DCT domain [29]. DCT seems more appropriate than the Fourier transform. Indeed, most video coding schemes are based on a DCT of the residual signal, and it is therefore more coherent to use the same transform for motion estimation and for the prediction error coding. However, such an approach faces the same difficulties as the Fourier representation.

### 5.02.5.2 Motion estimation in the wavelet domain

The wavelet transform enables the **multi-resolution** analysis of a signal using a critically sampled filterbank. We now discuss in more details the case of motion estimation in the wavelet domain.

#### 5.02.5.2.1 Problems raised by subsampling

To consider motion estimation on the coefficients of each wavelet subband, the lack of translation invariance of wavelet decompositions has to be taken into account. More specifically, the major problem with a **dyadic** subband decomposition is the fact that the subsampling operation applied after each filtering operation does not preserve the translation invariance.

Let us first consider the **redundant wavelet transform** of a signal  $f(t)$ , which is given by [30]:

$$\tilde{c}_j^\vartheta[f] = \int_{-\infty}^{\infty} f(t) \frac{1}{2^{j/2}} \Psi\left(\frac{t - \vartheta}{2^j}\right) dt, \quad (2.40)$$

where  $\Psi$  is the mother wavelet,  $\vartheta$  is the translation parameter, and  $j$  is the scale parameter. The **redundant decomposition** of the same signal translated by a factor  $\tau$ ,  $f^\tau(t) = f(t - \tau)$  is given by

$$\tilde{c}_j^\vartheta[f^\tau] = \int_{-\infty}^{\infty} f(t) \frac{1}{2^{j/2}} \Psi\left(\frac{t - (\vartheta - \tau)}{2^j}\right) dt = \tilde{c}_j^{\vartheta - \tau}[f]. \quad (2.41)$$

It can be observed that such a **redundant** transform is **translation invariant**. In other words, if the input signal is translated by the factor  $\tau$ , the transform coefficients also translate by the same amount.

On the contrary, a **non-redundant decomposition** lacks this desirable property. The decimated, thus non-redundant, wavelet coefficients of the input signal  $f(t)$  are now given by

$$c_j[k] = \int_{-\infty}^{\infty} f(t) \frac{1}{2^{j/2}} \Psi\left(\frac{t}{2^j} - k\right) dt, \quad (2.42)$$

对应关系

缺点

拓展DCT

冗余对  
平移不变性的  
影响

while the coefficients of  $f^\tau(t)$  are given by

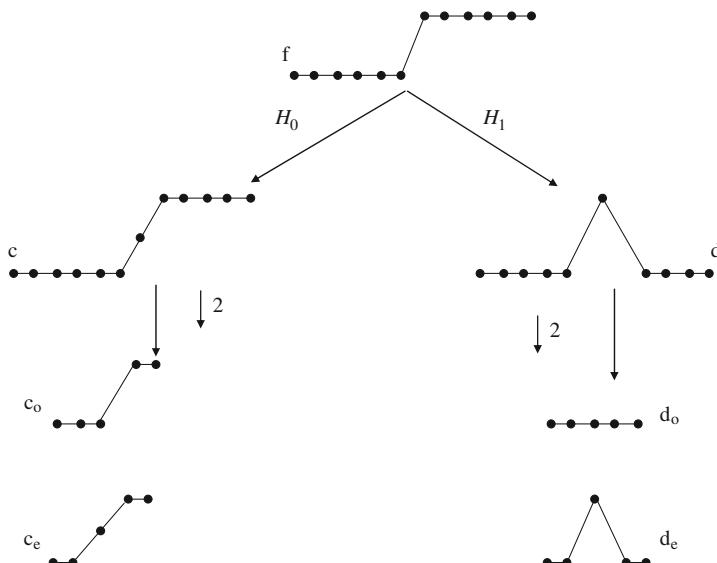
$$\begin{aligned} c_{j,\tau}[k] &= \int_{-\infty}^{\infty} f(t - \tau) \frac{1}{2^{j/2}} \Psi\left(\frac{t}{2^j} - k\right) dt \\ &= \int_{-\infty}^{\infty} f(t) \frac{1}{2^{j/2}} \Psi\left(\frac{t}{2^j} - \left(k - \frac{\tau}{2^j}\right)\right) dt. \end{aligned} \quad (2.43)$$

The translation invariance is preserved only when  $\tau$  is a multiple of  $2^j$ , at each decomposition level  $j$ . Otherwise, the translation invariance does no longer hold.

### 5.02.5.2.2 Motion estimation of the subsampled subbands

As a consequence of the above observations, the subsampling operation generates completely different subband decompositions, depending on the **position of the edges** of a moving object with respect to the **dyadic grid** (given by the multiples of  $2^j$  at each decomposition level). An illustrative example is given in [Figure 2.5](#). The subsampling operation will keep either the odd, or the even samples. Thus, a moving edge will result in a high frequency coefficient in the first case, and it will not appear at all in the second case. Let us analyze in more detail this example.

Suppose that Haar filters are used, i.e.,  $H_0$  and  $H_1$  are respectively the two-tap low and high-pass filters. By filtering, the unique signals  $c$  and  $d$  are obtained. However, the downsampling has not a unique result, and it depends on whether odd or even pixels are retained, that is  $c_o$  and  $d_o$ , respectively  $c_e$  and  $d_e$ , are obtained as approximation, respectively detail coefficients. If the initial step function



**FIGURE 2.5**

Odd and even downsampling of a step function.

不同结果  
影响因素

is moving from instant  $T$  to instant  $T + 1$  one pixel to the left and the odd pixels are retained, then  $c_o$  and  $d_o$  are the subbands of the step function at time  $T$ , while  $c_e$  and  $d_e$  are the subbands of the function at time  $T + 1$ . Moreover, we see that the prediction of the low frequency subband  $c_e$  from  $c_o$  is difficult, but maybe still possible in this low frequency subband. On the other hand, prediction of the high frequency subband  $d_e$  from  $d_o$  is impossible. It is however always possible to predict either the subsampled subbands ( $c_o$  and  $c_e$ , respectively,  $d_o$  and  $d_e$ ) from the corresponding non-subsampled one ( $c$ , respectively  $d$ ). In other words, one cannot estimate motion directly from the subsampled subband coefficients. Motion prediction has to take into account non-decimated coefficients.

Motion estimation in subsampled subbands at time  $T$  is possible if the unsubsamped subbands at time  $T - 1$  are known. Indeed, subsampling can be performed in four different ways, depending on whether the even or the odd samples in horizontal and vertical directions are retained. For an image, this yields four subsampled subbands at the first decomposition level. Interleaving these four subbands yields again the unsubsamped filtered image and thus no information is lost. Any of these subbands can be predicted from the unsubsamped one. The same procedure is iterated in order to determine the unsubsamped subbands at higher levels. The number of choices multiplies with the number of levels. For example, an image decomposed over 4 levels will lead to  $4^4 = 64$  possible ways of subsampling.

Any movement in the original field also appears in the unsubsamped subbands at any resolution level, because a time-invariant representation of the signal is used.

Therefore, a ME and MC procedure in subbands can be applied, estimating the motion vectors in the current subsampled subband from the previous unsubsamped subband [31]. As we have just seen, the procedure outlined above is more efficient than using ME/MC in unsubsamped subbands and then subsampling.

If the above algorithm is applied to each of the subbands of an image, then several motion vectors will be obtained for the same area in the image (as appearing at different resolutions and orientations). In order to ensure a coherence of the motion vector field over scales and, at the same scale, at different orientations, several strategies are possible. For example, one can consider joint optimization criteria for all the coefficients at a given resolution level and having the same spatial indices (these criteria depend on the chosen motion estimation algorithm, e.g., pel-recursive, block-matching, etc.). This will lead to the same motion vector at a certain scale. The coherence over scales can be achieved by propagating the motion vector obtained at coarse levels to finer levels: it can be used for initializing the motion estimation algorithm at that level. If a pel-recursive method is implemented, this is easily done in the first step of the algorithm. If a block matching method is under study, the initial vector can be used to restrict the search area and thus to accelerate the algorithm.

A similar procedure is applied in [32]. It is emphasized that performing downsampling after the selection of candidate blocks is crucial in order to ensure a full resolution motion estimate. However, the error criterion is evaluated after subsampling. Note that the motion search is applied to all subbands simultaneously, resulting in the same displacement vector for all subbands. This is realized by considering in the motion block e.g.,  $4 \times 4$  samples from each of the 16 subbands of a full-grown tree. Performing analysis before ME results in reducing the problems caused by the blocking artifacts.

Note that the motion estimation procedure presented in this section leads to a completely scalable video scheme, since the Displaced Frame Difference (DFD) is already decomposed into subbands. An embedded algorithm can thus code both DFD and the initial motion vector field.

### 5.02.5.2.3 Wavelet domain optical flow measurement

This method, proposed in [33], is based on the projection of the optical flow constraint (see [Section 5.02.2.3.1](#)) onto a wavelet basis. We remind the optical flow equation:

$$u(x, y) \frac{\partial f}{\partial x}(x, y, t) + v(x, y) \frac{\partial f}{\partial y}(x, y, t) + \frac{\partial f}{\partial t}(x, y, t) = 0. \quad (2.44)$$

The bi-dimensional wavelets can also be written, in a condensed form, as  $\Psi_{j,k}^s(x, y)$ , with  $s \in \{H, V, D\}$  (the three orientations of the details),  $j$  being the resolution level and  $k = (n, m)$  the spatial location where the inner product is computed. Using this notation, one can project the previous optic flow equation onto a wavelet basis, and get:

$$\begin{aligned} & \left\langle \Psi_{j,k}^s(x, y), u(x, y) \frac{\partial f}{\partial x}(x, y, t) \right\rangle + \left\langle \Psi_{j,k}^s(x, y), v(x, y) \frac{\partial f}{\partial y}(x, y, t) \right\rangle \\ & + \left\langle \Psi_{j,k}^s(x, y), \frac{\partial f}{\partial t}(x, y, t) \right\rangle = 0 \end{aligned} \quad (2.45)$$

for all  $s$ , and where the  $\langle \dots \rangle$  symbol represents the inner product.

One can also easily accept that the motion is constant over the support of the wavelets (these have been chosen with finite and short support). Thus, the previous equation becomes:

$$\begin{aligned} & -u(x, y) \left\langle \frac{\partial \Psi_{j,k}^s(x, y)}{\partial x}, f(x, y, t) \right\rangle - v(x, y) \left\langle \frac{\partial \Psi_{j,k}^s(x, y)}{\partial y}, f(x, y, t) \right\rangle \\ & + \left\langle \Psi_{j,k}^s(x, y), \frac{\partial f}{\partial t}(x, y, t) \right\rangle = 0. \end{aligned} \quad (2.46)$$

Here, the inner products are nothing else than the wavelet coefficients of the input image (therefore, they can be computed with fast algorithms). The last term (involving a temporal derivative) can be estimated with finite time differences of the wavelet coefficients of the image.

As this equation can be written at all scales  $j$  and for all orientations  $s$ , we have an overdetermined system of equations allowing the estimation of the two components of the motion in a robust manner. This overdetermined system can be solved by least mean squares at each scale of observation. The algorithm proceeds from coarse-to-fine scales: for each location  $(x, y)$ , the coarse estimation takes into account (by the inner products) the information in a window roughly given by the support of the wavelet function at that scale. The fact that from a given location to an adjacent one the support of wavelets overlaps ensures smooth flow estimation. The coherence of the final motion field is also guaranteed by the way information coming from different scales is combined into a unique motion vector. Similar approaches have been proposed in [34–36].

The motion vector obtained at coarse resolution is propagated at the finer scale and decomposed in two parts: one corresponding to an integer displacement at the coarse scale grid, which remains as it is, and the other, which cannot be estimated with good accuracy from the coarse grid and will be refined by solving the system at finer scales.

Another aspect that has been highlighted in [37] is the fact that using *real* wavelets for estimation is not optimal. Instead, *analytic* wavelets are proposed and shown by experiments to lead to more stable

results. The explanation would come from the fact that real wavelets act more or less like damped “cosine” functions. When solving the equations for optical flow, dividing by such oscillating functions can lead to division by zero, which, of course, provides a very unstable result. Analytic wavelets would act more like complex exponential functions, whose modulus does not cancel. Therefore, a denominator involving such functions would be more stable. Design of fast algorithms for dyadic decompositions with analytic wavelets can be found in [33]. However, a drawback of this approach is also the use of analytic wavelets, as they are not used for spatial analysis: this constrains to design a specific basis only for motion estimation (that cannot be simultaneously used for coding the prediction error).

## 5.02.6 Block matching approaches

Block matching methods have been specifically developed in the framework of image sequence coding. All video coding standards to date, including H.264/AVC [38] and HEVC [39, 40], are based on this paradigm. This class of techniques is therefore more thoroughly discussed in this section.

问题本质

Block matching methods enter the category of **matching-primitive techniques**. The aim is to **minimize a dissimilarity measure**. In particular, the fact that the same motion vector is estimated over an entire block can be seen as an additional **“smoothness” constraint** for solving the motion equation. In this way, the under-determined system can be solved. In addition, these methods will be quite **robust** to noise, which is not always the case with methods providing dense motion fields, such as the global optimization and the pel-recursive techniques.

Let us introduce some notation. Let us consider images of size  $N \times M$ . A block  $B_{p,q}$  is a set of indexes defined starting from  $(p, q)$  and whose size is  $P \times Q$ :

$$B_{p,q} = \{p, p+1, \dots, p+P-1\} \times \{q, q+1, \dots, q+Q-1\}. \quad (2.47)$$

The current image is divided into non-overlapping rectangular blocks as shown in [Figure 2.6](#). The typical values for  $P$  and  $Q$  are 4, 8, or 16, but larger block sizes are possible in the new standard HEVC [40].

In any case, for all the pixels in the block, a **single motion vector** is computed. With a small abuse of notation, we will refer to the vector of image intensity values within the block as  $f_k(B_{p,q})$ :

$$f_k(B_{p,q}) = [f(p, q, k), f(p+1, q, k), \dots, f(p+P-1, q, k), \dots, f(p+P-1, q+Q-1, k)]^T. \quad (2.48)$$

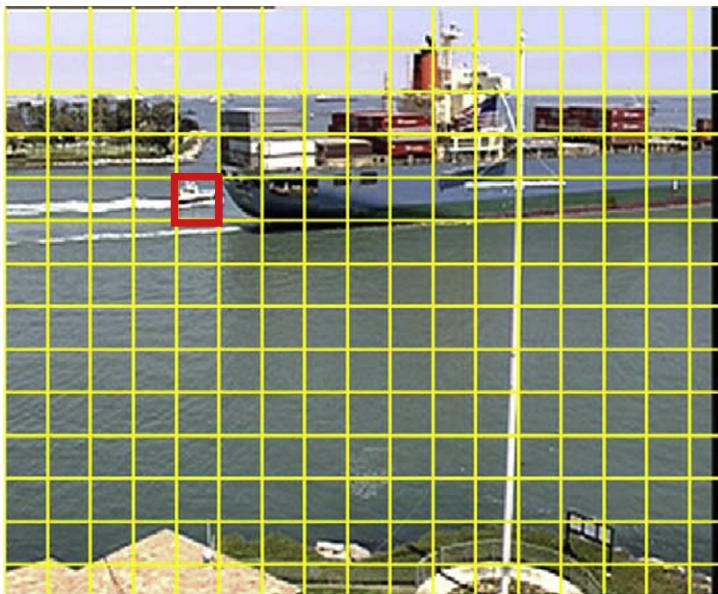
Block matching motion estimation is performed by computing a similarity measure between  $f_k(B_{p,q})$  and  $f_h(B_{p-i,q-j})$ , i.e., by comparing the intensity values in a block of the current image  $f_k$  and those in a block of a second image  $f_h$ , called **reference**. This second block is displaced from the initial position  $(p, q)$  by a vector  $D = (i, j)$  that is called **candidate** motion vector.

The case where  $h = k - l$ , with  $l > 0$ , (i.e., when we look for the displacement of blocks in the previous images) is called **forward motion estimation** and is shown in [Figure 2.7](#).

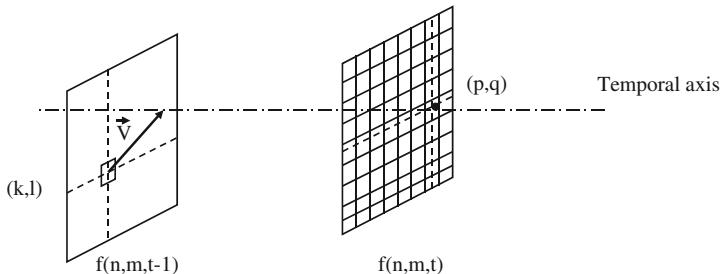
特例定义

The candidate vector that minimizes a suitable similarity measure between the blocks is the estimated vector for the block  $B_{p,q}$ :

$$(\hat{i}, \hat{j}) = \arg \min_{(i,j) \in W} d[f_k(B_{p,q}), f_h(B_{p-i,q-j})], \quad (2.49)$$

**FIGURE 2.6**

An image from the *container* sequence, divided into non-overlapping blocks of  $32 \times 32$  pixels. A single block  $B_{p,q}$  is highlighted.

**FIGURE 2.7**

Block matching algorithm with forward motion estimation.

where  $W$  is a suitable set of candidate vectors, commonly called search window. For a given block, the function to be minimized in Eq. (2.49) depends only on the candidate vector, so we will shorten it as  $J(i, j)$ . As a consequence, Eq. (2.49) becomes:

$$\left( \hat{i}, \hat{j} \right) = \arg \min_{(i,j) \in W} J(i, j). \quad (2.50)$$

变形，衍生

**Equations (2.49)** and **(2.50)** contain the essence of the method. The **various versions** of block matching differ among them for:

1. The **search strategy**, i.e., how the search window is scanned in order to find the best vector.
2. The **matching criterion**, that is the function  $d(\cdot, \cdot)$  used in [Eq. \(2.49\)](#).
3. The **block size** and **shape**.

These elements are addressed in the next subsections, which will be concluded by considering some other issues such as sub-pixel accuracy and recent advances.

定义  
形状

### 5.02.6.1 Search window and search strategy

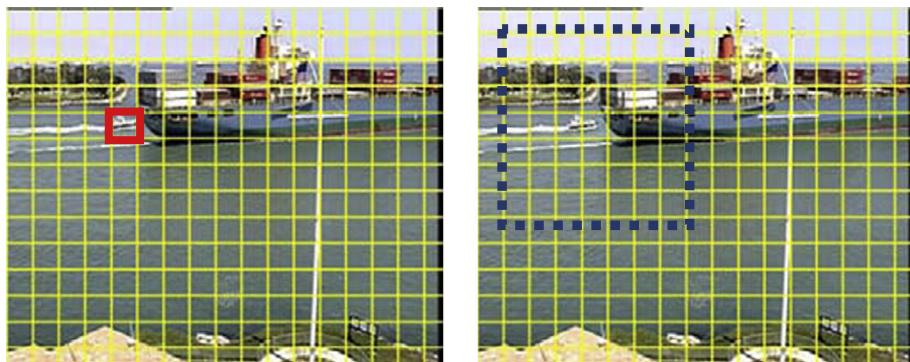
The motion vector is selected within a suitable set of candidate vectors. This **set** is called *search window* and represents the area where the most similar block will be searched. In the most common case, the search window corresponds to a rectangular area centered in the block  $B_{p,q}$  of the reference image. An example of a search window is shown in [Figure 2.8](#).

The **structure** of the search window has a huge impact on both the **complexity** of the motion estimation algorithm and on its **precision**. Therefore the choices of the search window and of the associated search strategy are critical for a motion estimation algorithm.

#### 5.02.6.1.1 Full search

A very intuitive choice for the search area is the whole reference image: the block  $B_{p,q}$  is compared to **all** possible blocks of the reference image. However this approach is extremely complex, since the number of candidate blocks is  $(N - P) \times (M - Q) \approx NM$ , and for each candidate block we have to compute the similarity measure.

However, it is not necessary to consider all pixels in the reference image: according to the characteristics of the movement and to the resolution of the image, typically it suffices to consider a rectangular area centered around the position  $(p, q)$ .



**FIGURE 2.8**

Left: The current image with the block  $B_{p,q}$  highlighted. Right: The reference image. The search window centered in  $(p, q)$  is shown.

Formally, the search window  $W$  is defined as a set of vectors:

$$W = \{-A, \dots, -1, 0, 1, \dots, A\} \times \{-B, \dots, -1, 0, 1, \dots, B\}. \quad (2.51)$$

The horizontal and vertical sizes of the search area can be different, according to the fact that horizontal movements are usually wider than vertical ones in natural videos. However, for the sake of simplicity,  $A$  and  $B$  have very often the same value, as we will assume in the following. Also, we refer to  $n = 2A + 1$  as the *width* of the search window.

In order to solve Eq. (2.49) or (2.50) with this definition of  $W$ , we need to compute the criterion for  $n^2$  candidates: this approach is called **full search**. Full search assures to find the **best motion** vector (i.e., the one minimizing the criterion) within all displacements of no more than  $A$  pixels in whichever direction, but has a relatively **large complexity**, proportional to  $n^2$ . However, this technique lends itself to a parallel implementation since the estimation can be performed independently for each block of the current image.

**Other techniques** exist that can determine a motion vector with a smaller complexity. However these are **suboptimal**, in the sense that they cannot assure that the best vector among all those having components bounded by  $A$  will be found. The most popular sub-optimal search strategies are described in the following subsections.

#### 5.02.6.1.2 Fast search: definitions

In order to reduce the computational complexity of the full-search method presented above, sub-optimal techniques have been proposed. The basic idea is to select a subset of  $\{-A, \dots, A\}^2$  as the search window. In this way, we still may estimate large movements, but with less than  $n^2$  computations of the criterion.

Fast search techniques are generally **iterative**. We start with a search window made up of a few and relatively large vectors, and with an initial candidate vector  $(i_0, j_0)$ , e.g.,  $(0, 0)$ .

At the  $k$ th step, the current best vector  $(i_k, j_k)$  is selected by full search in the current search window  $W_k$ . Then, the **search window is modified**: typically it is centered on  $(i_k, j_k)$  and possibly scaled. A proper **stopping condition** is needed: for example, a maximum number of iterations or a condition on the elements of  $W_k$ .

The general structure of the algorithm is the following:

1. Initialization: we set  $k = 0$ , and we choose  $W_1$  and  $(i_0, j_0)$ .
2. While the stop condition is not met,
  - a.  $k = k + 1$ ,
  - b.  $(i_k, j_k) = \arg \min_{i, j \in W_k} J(i, j)$ ,
  - c.  $W_{k+1} = \phi [W_k, (i_k, j_k)]$ .
3.  $(\hat{i}, \hat{j}) = (i_k, j_k)$ .

Of course, the key feature of this algorithm is the **modification of the search window at each step**, represented by the function  $\phi$ . We also notice that, if  $K$  iterations are performed, the number of criterion computations is

$$C' = \sum_{k=1}^K \text{card}(W_k). \quad (2.52)$$

### 5.02.6.1.3 2D-logarithmic search

In the 2D-logarithmic search [41], we start with a rough search grid, and we refine it when the current estimated vector is the center of the current search area.

For this case, we need to define the following sets of indexes:

$$\Gamma(1) = \{-1, 0, 1\}^2, \quad (2.53)$$

$$\Gamma(2^j) = \{(0, 0), (\pm 2^j, 0), (0, \pm 2^j)\}, \forall j > 0. \quad (2.54)$$

We start with a given  $(i_0, j_0)$ , for example the null vector. We select the initial search area  $W_1 = \Gamma(2^m)$ . Typically,  $m = 3$  or  $4$ . The value  $r_k = 2^m$  is the current **radius** of the search area. The general step of the algorithm is as follows. If the radius is not 1, the matching criterion is computed for the five candidates of the current window. If the best  $(i_k, j_k)$  candidate is the central one, we **halve** the search radius and set

$$W_{k+1} = \left\{ (i_k, j_k) + (i, j) | (i, j) \in \Gamma\left(\frac{r^k}{2}\right) \right\}. \quad (2.55)$$

Otherwise we just center the search window on the best candidate, and keep the same radius for the search area:

$$W_{k+1} = \left\{ (i_k, j_k) + (i, j) | (i, j) \in \Gamma(r^k) \right\}. \quad (2.56)$$

The search patterns are shown in [Figure 2.9](#).

Finally, if the search radius is equal to 1, the matching criterion is computed on the nine candidates of the current window, which is formed by the previous best candidate and its eight neighbors.

One of the main **drawbacks** of this algorithm is that the number of iterations is variable and therefore, the complexity of the whole procedure is not perfectly managed. Such a method is therefore **not suitable for hardware implementations**.

### 5.02.6.1.4 Conjugate directions search

The basic idea of this method is to perform **one-dimensional searches at each iteration**. We start by a **searching window** of the form

$$W_{k+1} = \{(i_k, j_k) + (i, j) | (i, j) \in \Delta W_k\}, \quad (2.57)$$

with

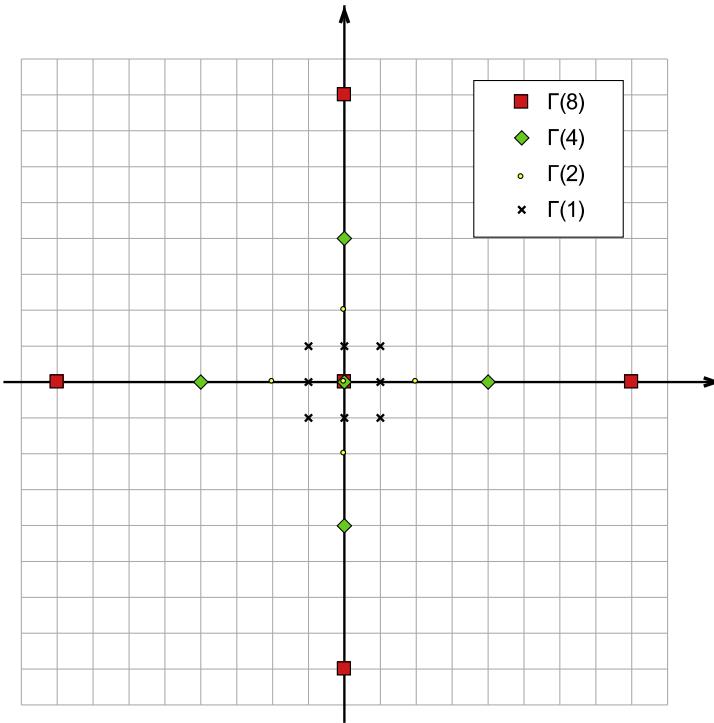
$$\Delta W_k = \{(-1, 0), (0, 0), (1, 0)\}. \quad (2.58)$$

条件

This is performed for the horizontal search, and we continue in this direction as long as the **current best vector is different from the center of the window and we do not reach a picture boundary**. When one of these two cases arises, we switch to the vertical search, and set the searching window to

$$\Delta W_k = \{(0, -1), (0, 0), (0, 1)\}. \quad (2.59)$$

When again a zero motion vector is found, either the estimation is **stopped**, or a third step can be envisaged, involving a search in the direction given by the original estimation point and the previous

**FIGURE 2.9**

2D-Log: search patterns at different steps.

estimation. Note however that this step is optional and it can be skipped if a limited-complexity algorithm is targeted.

A similar algorithm is the **cross-search** [42], where the search is performed separately in each dimension. In this case however, a full exploration of each direction is performed. In other words, we have only two steps, with

$$\Delta W_1 = \{-A, -A + 1, \dots, A\} \times \{0\}, \quad (2.60)$$

and

$$\Delta W_2 = \{0\} \times \{-A, -A + 1, \dots, A\}. \quad (2.61)$$

The cross-search algorithm requires the computation of the criterion  $2n - 1$  times instead of  $n^2$  of the full search.

#### 5.02.6.1.5 Three-step search

As the name shows, the aim of this algorithm is to obtain **uniform complexity**, corresponding to 3 iterations, regardless of the motion activity [43]. This can be achieved, of course, only at the expense of a certain loss in the precision of the motion vector.

The three-step search (**TSS**) algorithm can be seen as a variant of the 2D-logarithmic search with two differences:

- (1) The search window is always composed by nine elements:

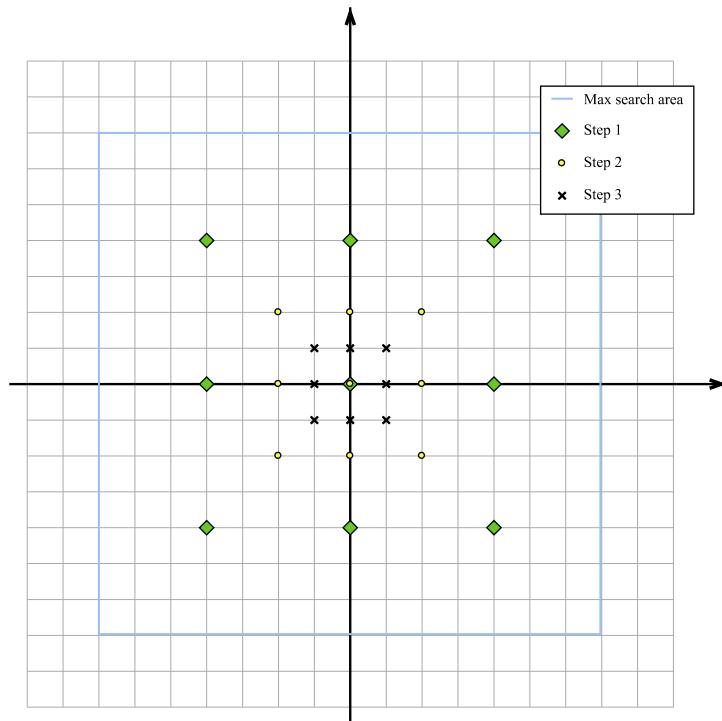
$$W_{k+1} = \{(i_k, j_k) + (i, j) | (i, j) \in \Delta W_k\},$$

with

$$\Delta W_k = \{-2^k, 0, 2^k\}^2.$$

- (2) The search radius is halved at each step independently from the estimated vector.

Therefore, if at the first iteration  $\Delta W_1 = \{-4, 0, 4\}^2$ , the algorithm will always stop after three steps. The corresponding search patterns are shown in [Figure 2.10](#). The TSS algorithm allows finding **displacements** of  **$\pm 7$  pixels** in both directions with only 25 computations of the criterion (nine for the first step and eight for both the second and the third, since the value of  $J$  for the center of the search window has been



**FIGURE 2.10**

Three-step search: search patterns.

computed at the previous step). For comparison, the full search would have required 225 computations, but would also have provided the best vector.

改进

Some variations of the TSS have been proposed in the literature, such as the new TSS [44] which is biased toward center points of the search window and allows an early termination of the algorithm, or the four-step search [45], which reduces the complexity by checking only a subset of  $\Delta W_k$ .

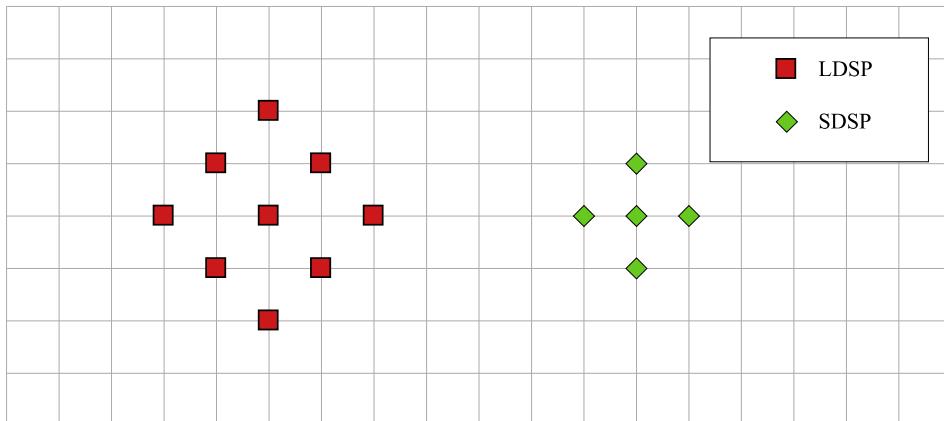
#### 5.02.6.1.6 Diamond search

The TSS and its variations used to be quite popular until the years 2000s, when a new generation of fast **block matching algorithms** have been developed. These algorithms are not limited to a  $15 \times 15$  search area but still are very fast and effective, and therefore are very commonly used in practical applications such as video encoders.

步骤

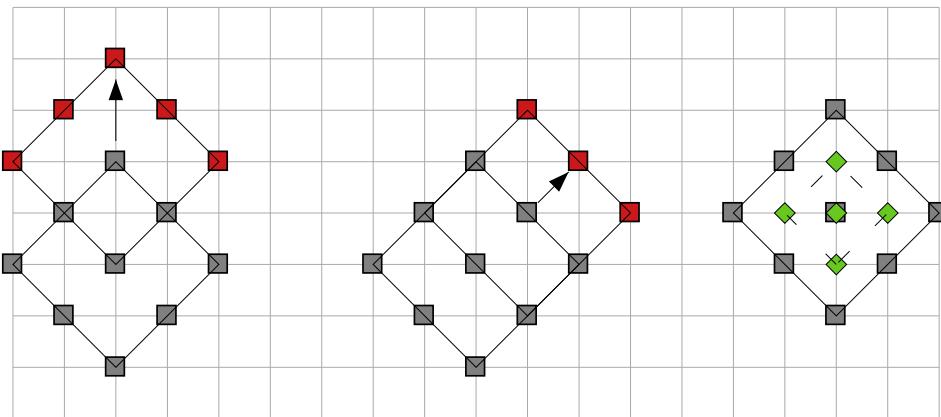
The **diamond search (DS)** [46] algorithm employs two search patterns as shown in Figure 2.11. At the first step, a large pattern with a diamond shape is selected (**LDSP** in Figure 2.11). According to fact that the **best point is the center or not**, the next search pattern is respectively changed into a small diamond-shape-pattern (**SDSP**) or is still a LDSP, as shown in Figure 2.12. Moreover, only a subset of the pattern points has to be checked at any new step, since the new pattern is always partially superposed to the old one. The algorithm **stops** when the best point of the SDSP has been found. In other words, in the DS, the LDSP moves in the reference image until the best position is the center of the pattern; then a last iteration of the algorithm is performed using the SDSP.

The DS algorithm offers very good performance since its **search pattern** is neither too small (which could trap the search algorithm into local minima) nor too large (such as the first pattern of the TSS, which could mislead the search path toward a wrong direction). Moreover, the **search area** is not limited since the search pattern keeps moving until the central position of the LDSP is found.



**FIGURE 2.11**

Large (LDSP) and small (SDSP) diamond search pattern.

**FIGURE 2.12**

The pattern for the next step is different according to the fact that a corner point (left), a side point (center), or the central point (right) is selected.

Simulation experiments show that DS largely outperforms TSS in terms of motion estimation quality, achieving performances similar to NTSS but with a complexity reduction of more than 20%. Thanks to the effectiveness of this technique, it has been integrated in the reference software of the MPEG-4 video coding standard.

#### 5.02.6.1.7 Hexagon search

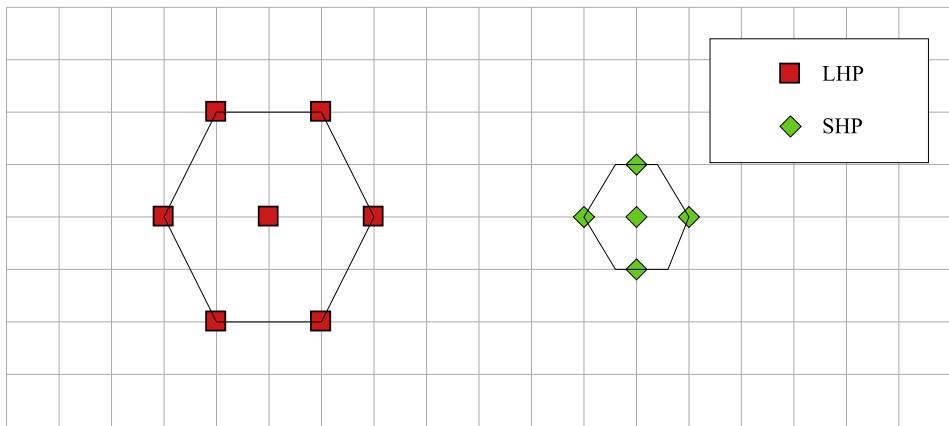
The DS algorithm has proven that pattern shapes other than a square can lead to fast motion estimation algorithms. However, as one can see from Figure 2.12, the LDSP moves with a speed of 2 pixels per iteration along the vertical and horizontal directions, while it moves at only  $\sqrt{2}$  pixel per iteration in the diagonal directions. This means that the algorithm could require too many iterations to find a diagonal motion vector. Therefore, a hexagonal search algorithm (HS) has been proposed [47], with the patterns shown in Figure 2.13. The patterns move according to the scheme shown in Figure 2.14. As in the DS, when the central point of the large pattern is chosen, the algorithm performs a last iteration using the small pattern. The hexagonal search has two advantages with respect to the DS: first, regardless of pattern motion, only three new points are tested per iteration; second, the pattern is more isotropic than that of DS, and so diagonal directions are not penalized with respect to horizontal ones. As a consequence, the HS achieves the same motion estimation precision as DS with a complexity reduction close to 40%. The HS (and some variations of it) are integrated into the reference software of the H.264/MPEG-4 AVC standard [38].

#### 5.02.6.2 Matching criterion

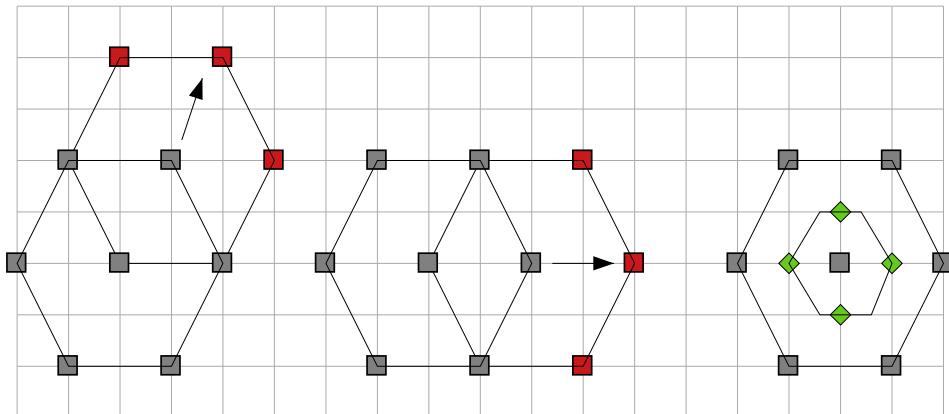
In this section we discuss some of the most popular matching criteria for block-based motion estimation.

场景

发现问题，什么情况下性能不好，提高整体性能

**FIGURE 2.13**

Large (LHP) and small (SHP) hexagonal patterns.

**FIGURE 2.14**

The movement of the hexagonal pattern: the speed is two pixels per iteration in the horizontal direction and  $\sqrt{5} \approx 2.236$  pixels per iteration in the diagonal directions.

#### 5.02.6.2.1 Norm-based criteria

As shown in Eq. (2.49) or (2.50), the blocks in the current and in the reference image are matched according to a suitable metric measuring their dissimilarity:

$$J(i, j) = d [f_k(B_{p,q}), f_h(B_{p-i,q-j})]. \quad (2.62)$$

The most natural approach is to use some distance between the vectors  $f_k(B_{p,q})$  and  $f_h(B_{p-i,q-j})$ , i.e., to compute the  **$p$ -norm** of their difference:

$$J(i, j) = \|f_k(B_{p,q}) - f_h(B_{p-i,q-j})\|_p^p. \quad (2.63)$$

Note that we raise the  $p$ -norm to the power  $p$  in order to get rid of the irrelevant  $p$ -order root. If we set  $p = 2$ , the block matching is performed minimizing the Euclidean distance between the vectors of image samples. Thus we have:

$$J_{SSD}(i, j) = \sum_{(n,m) \in B_{p,q}} [f(n, m, k) - f(n - i, m - j, h)]^2. \quad (2.64)$$

This matching criterion is referred to as the **Sum of Squared Differences (SSD)**. Note that using the SSD is equivalent to using the **mean square error (MSE)** between the current image and the compensated reference. The SSD is very **popular**, above all for compression applications for several reasons. First, the Euclidean distance is a very **intuitive and easy-to-understand** metric. Second, by minimizing the SSD, we minimize the mean square error between the current block  $f_k(B_{p,q})$  and the block  $f_h(B_{p-i,q-j})$ , which is used as a reference for prediction. Hence, this **allows an efficient residual coding**. In fact, the **Motion Compensated (MC) prediction** of the current image, i.e., a prediction of  $f_k$  where  $f_k(B_{p,q})$  is replaced by  $f_h(B_{p-i,q-j})$ , is often used to measure the **quality of the motion estimation**. By definition, no other criterion can provide a better result in this respect (for a given block size and search area). However, the SSD has some **drawbacks**. First, it is relatively **complex**, since it requires  $PQ$  multiplications to be computed (one per each pixel of the block). Second, if some pixels in the image are affected by **noise**, the square power tends to enhance the associated error, preventing to find the correct motion vector. Third, it does not take into account possible **global illumination variations** from one image to another.

In order to alleviate the first two problems, one can resort to the **sum of absolute differences (SAD)**, defined as follows:

$$J_{SAD}(i, j) = \sum_{(n,m) \in B_{p,q}} |f(n, m, k) - f(n - i, m - j, h)|. \quad (2.65)$$

The SAD is equivalent to the **Mean Absolute Error (MAE)**. In order to compare the SSD and SAD, we show in [Figure 2.15](#) two images from a video sequence and in [Figure 2.16](#) the motion vector fields obtained using block matching with the SSD (left) and the SAD (right) criteria. The two criteria have very close **qualities**: the MSE of the compensated image achieved by minimizing the SSD is only 0.16 dB smaller than the one obtained using SAD. We also observe that the two fields capture the global lateral motion of the sequence and the different apparent velocity of the foreground (the tree) and the background. However both methods fail at correctly estimating the movement of very homogenous areas (such as the sky in the left part of the image). In particular SSD seems to produce more outliers such as the vectors in red.

This lack of **regularity** affects both the capability of the ME to represent the real motion and the compression performances that can be achieved: an irregular MVF is more expensive to encode than a regular one. For example, the MVF estimated by SSD has an estimated coding cost of 2143 bits, while the slightly more regular field produced by SAD costs 2103 bits.

特例

优势

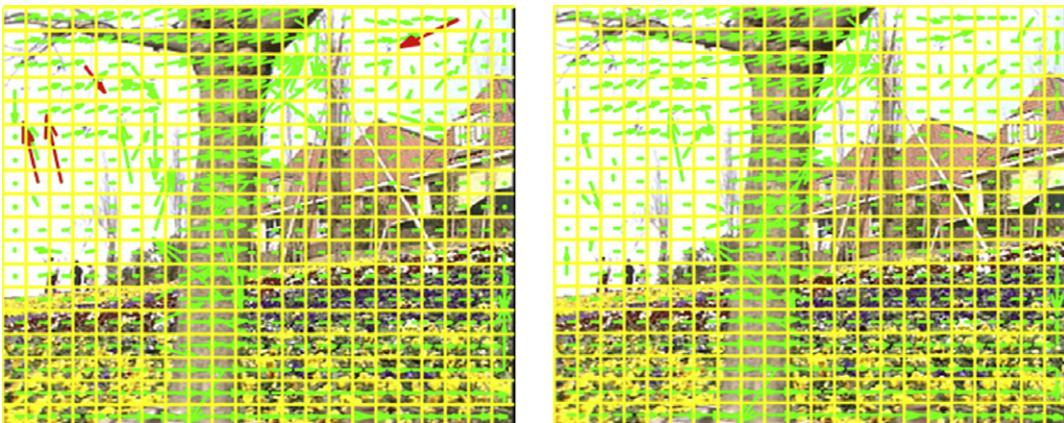
作用

劣势

两者比较

**FIGURE 2.15**

Two images (223 and 227 respectively) from the test sequence *flower and garden*.

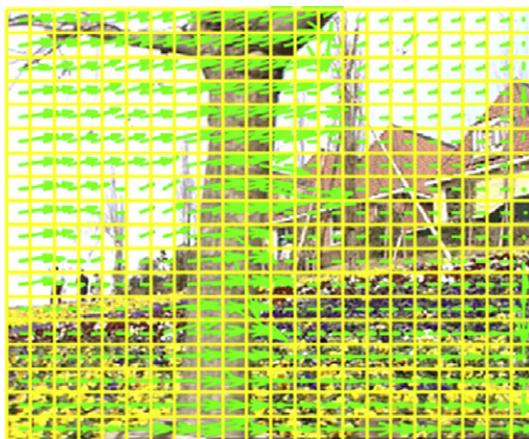
**FIGURE 2.16**

Estimated motion vector fields. Left: SSD criterion; right: SAD criterion.

**规律性 , 整齐** We can **improve** the regularity of the MVF by explicitly introducing a **smoothness constraint** in the criterion:

$$J_{REG}(i, j) = \|f_k(B_{p,q}) - f_h(B_{p-i,q-j})\|_p^p + \lambda R(i, j), \quad (2.66)$$

where  $R(i, j)$  is a suitable cost function and  $\lambda$  is a positive constant. As a consequence, the vector that has the smallest SSD or SAD is not selected if it is too irregular according to  $R$ . For example,  $R$  could be the norm of the difference between  $(i, j)$  and a representative of its neighborhood: this would allow

**FIGURE 2.17**

Estimated MVF using a regularized SSD criterion.

having a vector much different from its neighbors only when this is related to a new object (involving a large SSD or SAD reduction). In video compression applications,  $R(i,j)$  is the coding cost of the vector  $(i,j)$ , as we shall describe in more details in [Section 5.02.9](#).

An example of regularized MVF is shown in [Figure 2.17](#). We used the criterion in [Eq. \(2.66\)](#), with  $p = 2$ . The  $R$  function is the norm of the difference between  $(i,j)$  and the median vector of its causal neighborhood (three vectors). The resulting MVF is visually more regular than those shown in [Figure 2.16](#). This is confirmed by the estimated coding cost, which shrinks to 2008 bits, while the MC prediction quality is reduced by less than 0.1 dB.

### 5.02.6.2.2 An illumination-invariant criterion

Block-based matching algorithms are mainly used in compression; however, given their simplicity, they have been investigated in the context of other applications, where the objective is to find a MVF as close as possible to the actual motion. However, in these cases SSD and SAD show another limit: they cannot

假设：整个区域变化相同 deal with global illumination changes. In this case, we can resort to a variant of SSD which is robust to affine luminance transformations: the **Zero-mean Normalized SSD (ZN-SSD)**. If we refer to the  $h$ th element of vector  $f_k(B_{p,q})$  as  $f_k(B_{p,q})[h]$ , we can introduce the zero-mean version of vector  $f_k(B_{p,q})$  component by component:

$$\tilde{f}_k(B_{p,q})[l] = f_k(B_{p,q})[l] - \frac{1}{PQ} \sum_h f_k(B_{p,q})[h]. \quad (2.67)$$

We define then the ZN-SSD:

$$J_{\text{ZN-SSD}}(i, j) = \frac{\sum_{l=1}^{PQ} \left[ \tilde{f}_k(B_{p,q})[l] - \tilde{f}_h(B_{p-i,q-j})[l] \right]^2}{\left( \sum_{l=1}^{PQ} \tilde{f}_k^2(B_{p,q})[l] \times \sum_{l=1}^{PQ} \tilde{f}_h^2(B_{p-i,q-j})[l] \right)^{1/2}}. \quad (2.68)$$

The main **disadvantage** of ZN-SSD is its **computational complexity**: for each candidate vector we need to perform about  $3PQ$  multiplications.

### 5.02.6.2.3 Correlation-based criteria

来源

It is known that the **cross-correlation** (i.e., the **scalar product**) between two vectors is a measure of their **similarity**, and therefore one might think about using correlation-based criteria to perform block matching. This is also motivated by the fact that fast algorithms exist to compute the correlation in the **frequency domain**. However, as we show in the following, the cross-correlation itself has some major drawbacks and cannot reliably be used as matching criterion, while the normalized cross-correlation can effectively perform this task.

Let us start by the relationship between norm-based and correlation-based criteria. The SSD between  $f_k(B_{p,q})$  and  $f_h(B_{p-i,q-j})$  can be written as:

$$\begin{aligned} J_{\text{SSD}}(i, j) &= \|f_k(B_{p,q}) - f_h(B_{p-i,q-j})\|^2 \\ &= \langle f_k(B_{p,q}) - f_h(B_{p-i,q-j}), f_k(B_{p,q}) - f_h(B_{p-i,q-j}) \rangle \\ &= \|f_k(B_{p,q})\|^2 - 2\langle f_k(B_{p,q}), f_h(B_{p-i,q-j}) \rangle + \|f_h(B_{p-i,q-j})\|^2. \end{aligned} \quad (2.69)$$

假设

In this expression,  $\|f_k(B_{p,q})\|^2$  does not depend on  $(i, j)$ . If the **norm of the displaced block does not vary very much with  $(i, j)$** , then minimizing the SSD criterion is almost equivalent to maximizing the correlation between the blocks:

$$\begin{aligned} J_{\text{CORR}}(i, j) &= \langle f_k(B_{p,q}), f_h(B_{p-i,q-j}) \rangle \\ &= \sum_{(n,m) \in B_{p,q}} f(n, m, k) f(n - i, m - j, h). \end{aligned} \quad (2.70)$$

The main **advantage** in using this criterion is that fast FFT-based implementations exist for the correlation calculation. On the other hand, the cross-correlation is **not very reliable** for the following reasons:

- (1) In natural images, the blockwise energy  $\|f_h(B_{p-i,q-j})\|^2$  actually **varies** with the position. Therefore the correlation of the original block with the real displaced block can be less than the correlation with a very bright spot in the reference image.
- (2) The correlation coefficient is **sensitive** to **global amplitude changes**, such as those caused by changing global illumination conditions.

Therefore, a better criterion is the normalized correlation coefficient, defined as:

$$J_{N-\text{CORR}}(i, j) = \frac{\sum_{l=1}^{PQ} [\tilde{f}_k(B_{p,q})[l] \tilde{f}_h(B_{p-i,q-j})[l]]}{\left(\sum_{l=1}^{PQ} \tilde{f}_k^2(B_{p,q})[l] \times \sum_{l=1}^{PQ} \tilde{f}_h^2(B_{p-i,q-j})[l]\right)^{1/2}}. \quad (2.71)$$

We observe that Eq. (2.71) is very similar to Eq. (2.68): the only difference is that the zero-mean cross correlation has replaced the zero-mean SSD at the numerator. Removing the local mean and normalizing allows to mitigate the impact of the two abovementioned problems of cross correlation.

**用途**

The normalized cross-correlation is typically used for **feature tracking** in an image sequence: in this case we look for the position of a single feature, which is no longer constrained to have a rectangular support. Therefore, Eq. (2.71) is simply modified, considering the vector of luminance values of the template instead of  $f_k$  and a vector with the same shape (but displaced by  $(i, j)$ ) instead of  $f_h$ .

The main **reason** for using **normalized** cross-correlation instead of SSD is that, when the number of pixels of the feature is much smaller than the number of pixels of the image, frequency-domain implementations of Eq. (2.71) are very efficient in terms of **computational complexity**. They are mainly based on FFT for the computation of the numerator and the computation of a cumulated sum over the image for the denominator.

#### 5.02.6.2.4 A criterion for wavelet-based compression

Norm based criteria, such as SAD, SSD, and their regularized versions, prove to be very effective for predictive video compression, since in this framework motion estimation and compensation are used to **minimize the energy of the prediction error**. It is interesting to note that, when the **temporal correlation** of a video signal is **not removed by predictive coding but by transform coding**, as in the case of motion compensated wavelet video coding, these criteria are no longer necessarily optimal. In fact, instead of minimizing the prediction error energy, in this case one **should maximize the transform coding gain**, defined as the **ratio** of geometric and arithmetic mean of the temporal wavelet subbands. These subbands are obtained by wavelet filtering along the motion direction, which in turn is computed by motion estimation. Therefore, the motion estimation should find out the **trajectory** that maximizes the coding gain. This can be quite difficult in the general case, but for a class of simple yet effective temporal filters it has been found that optimal motion estimation should be performed on three consecutive frames, evaluating jointly the backward and forward motion vectors [48]. If  $\varepsilon_B[\varepsilon_F]$  is the backward [forward] motion compensated error, instead of separately computing the MVFs that minimize  $\|\varepsilon_B\|^2$  and  $\|\varepsilon_F\|^2$ , it has been proved that the optimal MVFs are those that jointly minimize  $\|\varepsilon_B\|^2 + \|\varepsilon_F\|^2 + \langle \varepsilon_B, \varepsilon_F \rangle$ . It is interesting to notice that this criterion involves the computation of norms and correlations.

#### 5.02.6.3 Sub-pixel accuracy

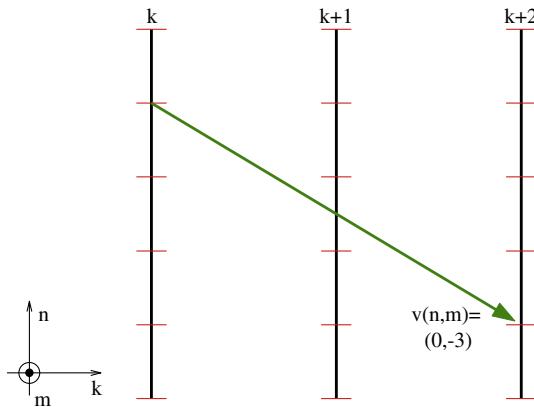
**描述**

Sub-pixel accuracy refers to displacements that do not correspond to integer pixel positions on the original image grid. The need to estimate finer displacements can be understood from the simple example in Figure 2.18. The horizontal dimension on the image plane is orthogonal to the drawing plane. We assume that the motion vector from image  $k$  to  $k + 2$  in the position  $(n, m)$ , referred to as  $v(n, m)$ , is estimated as  $(0, -3)$ , i.e., three pixels downward. Assuming a uniform motion in the corresponding time interval, one would infer a displacement of 1.5 pixels downward from frame  $k$  to frame  $k + 1$ .

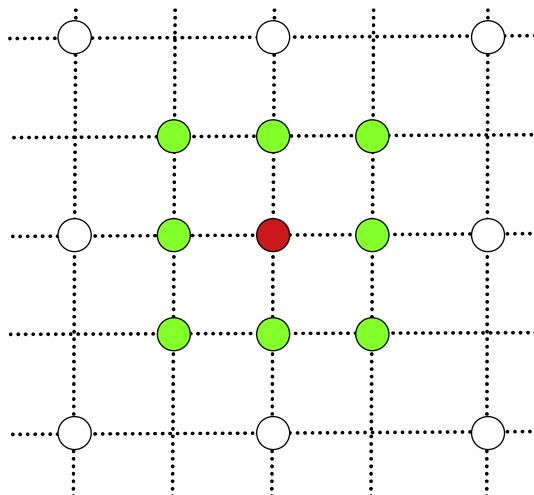
The estimation of motion vectors with fractional pixel accuracy is possible and it is frequently implemented in block matching algorithms. We describe below the method for the case of half-pixel accuracy, but it is easy to generalize it to other precisions ( $1/3$ ,  $1/4$ ,  $1/8$ , and so on). For a better understanding, the method is illustrated in Figure 2.19.

**步骤**

Once the best fitting for **integer pixel accuracy** has been found (the red pixel in Figure 2.19), one has to consider all the candidates at **fractional positions**  $(i, j)$  (in green) and to test the same criterion  $J(i, j)$  (MAE or MSE). The vector minimizing the criterion  $J(i, j)$  among these positions is the new motion vector, having fractional values. Of course, the difficulty consists in the fact that for testing the matching

**FIGURE 2.18**

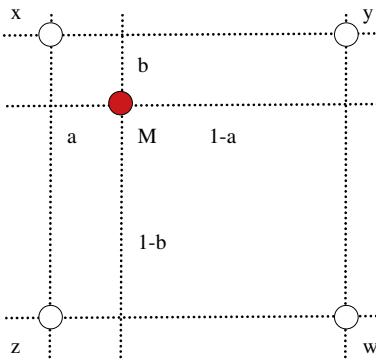
A schematic representation of the sub-pixel motion.

**FIGURE 2.19**

Half-pel accuracy motion estimation: the red position on the integer grid is the currently estimated motion vector, the white pixels are its neighbors on the integer grid, the eight green positions on the half-pel accuracy grid have to be tested. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this chapter.)

criterion, we do not dispose of image values at half-pel positions. These have to be interpolated from existing values. A common method used for this operation is bilinear interpolation, but longer filters could also be used.

Let us briefly restate the principle of the bilinear interpolation, with reference to [Figure 2.20](#).

**FIGURE 2.20**

Bilinear interpolation: white pixels are on the integer grid, red value has to be interpolated. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this chapter.)

具体计算  
技术细节

If  $x, y, z, w$  are the intensity values of the neighbors situated on the integer grid (consider the distance between them equal to 1) and the point to be interpolated is at distance  $a$  to the left-hand side neighbors and  $b$  to the upper neighbors, then the value  $M$  obtained with bilinear interpolation will be:

$$M = (1 - a)(1 - b)x + a(1 - b)y + b(1 - a)z + ab w. \quad (2.72)$$

In particular, for half-pel accuracy,  $a = b = 1/2$  and then  $M = (x + y + z + w)/4$ .

The concept of **generalized interpolation** is introduced in [49]. A combination of short Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters is used, which provides greater design flexibility and better coding performance. A hardware-friendly multiplication-less implementation is also described.

泛化  
发展

Sub-pixel motion estimation and compensation has been integrated in early video coding standards, and has been progressively improved since then. In MPEG-1 and MPEG-2 a simple bilinear interpolation allowed half-pixel motion precision. In MPEG-4 Part 2 an eight-tap filter is used to compute the interpolated samples.

In H.264/AVC [38], a quarter-pixel accuracy is allowed for motion vectors. A six-tap filter is used to generate half-pixel samples. These values are then rounded to integer values, on which bilinear filtering is performed to produce quarter-pixel samples. Interpolation-free methods for H.264/AVC sub-pixel accuracy have also been proposed [50], using the full pixel sum of absolute difference distribution of each block.

HEVC [39, 40] further improves the interpolation filters used for **fractional ME**. An eight-tap filter is used to generate half-pixel samples and a seven-tap one is used for quarter-pixel samples. HEVC takes benefit both from longer filters and from not having intermediate rounding operations.

#### 5.02.6.4 Variable-size block matching

限制

One severe **limitation** of block matching motion estimation algorithms is that a **single** translational motion vector is assigned to all pixels within a block. However, the underlying assumption that the

whole block is undergoing a uniform translational motion does not hold in the case of **complex** scenes with fast moving objects. In particular, it leads to a poor prediction along moving edges which results in block artifacts in the MC frame and decreased compression performance.

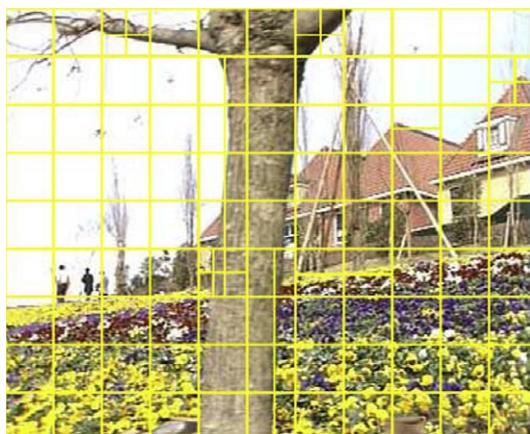
To alleviate this important drawback, **variable-size block matching** techniques, also referred to as **multi-grid**, have been proposed [3, 51]. These techniques are based on the observation that large blocks are sufficient in uniform areas, whereas finer blocks are necessary in highly textured regions or near moving object edges. Straightforwardly, in variable-size block matching techniques, the size of blocks is adapted based on **local texture** and **motion characteristics**.

### 选取原则

In practice, variable-size block matching techniques essentially proceed as follows. **Block matching** is first performed on a **coarse grid**. Then, selected blocks are **split**. For instance, this decision can be based on a simple threshold on the SAD or SSD of the block, to identify blocks where motion estimation has failed. Each of the new **sub-blocks** is assigned the **motion vector** of the **parent block**, and block matching is then performed again on these sub-blocks. Note that a smaller search window is most often used at this stage, as an initial estimate of the motion vector is already available. The process is iterated until a minimum block size is reached. At each step of the iterative process, selected blocks are most commonly divided into **two** or **four** sub-blocks, resulting in a binary- or quad-tree representation. This segmentation information can thus be efficiently represented. An example of the grid resulting from this process is shown in [Figure 2.21](#). Clearly, large blocks are used in uniform areas, whereas small blocks have been preferred in detailed areas.

The projection of the motion vector obtained at a coarse grid to a finer one should have two **objectives**. Firstly, the projection operator should avoid the propagation into finer levels of erroneous motion vector estimates due to large block sizes. Secondly, it should guarantee the **smoothness** of the motion vector field. Simply **duplicating** the parent vector may not be optimal. A **bilinear interpolation** is also possible. It typically leads to smoother motion fields; however, it does not ensure motion **consistency** along moving

### 一致性



**FIGURE 2.21**

Example of final grid using variable-size block matching.

object boundaries. A better approach is to select the best initial condition among motion vectors in a neighborhood, as proposed in [3].

Regarding the splitting criterion, in the context of video coding, a more appropriate and sophisticated algorithm is to optimize the decision in a rate-distortion sense. More precisely, in this way, the gain of a more accurate motion precision, leading to a reduced residual signal, is weighted against the cost of extra motion vectors to be transmitted. See [Section 5.02.9.1.3](#) for a more detailed discussion on rate-distortion optimization.

A simple form of variable-size block matching is supported in H.264/AVC [38] and HEVC [39,40], as described in more detail in [Section 5.02.9.1](#).

实现

Efficient hardware implementations of variable-size block matching have been proposed. In [52], a one-dimensional very large-scale integration architecture is introduced for full-search variable-size block matching. The SAD of a larger block is efficiently computed by re-using the results previously obtained for smaller sub-blocks. In [53], the impact of variable-size block matching in hardware architectures is first analyzed, and two new hardware architectures are then proposed.

结合，融合

Finally, variable-size or multi-grid block matching techniques can efficiently be combined with multi-resolution approaches, as described in [Section 5.02.8](#) in order to further improve performance.

## 5.02.6.5 Related techniques

### 5.02.6.5.1 3D Recursive Search

场景  
要求2  
避免

组成

作用  
结果

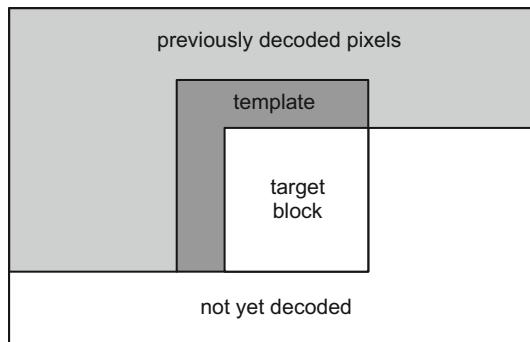
**3D Recursive Search** (3D RS) block matching [54] has been more specifically developed for frame rate conversion applications. In this context, a strong requirement is to generate a smooth motion vector field in order to prevent spurious distortion during motion compensated frame interpolation. A low complexity solution is another important constraint to enable efficient hardware implementation and integration into consumer television terminals.

In 3D RS, motion vectors are estimated recursively using block matching. Two spatial predictors, in two different directions, are used for better convergence performance. Furthermore, look-ahead temporal predictors are also used to accelerate convergence. An effective search strategy is presented in [54] based on asynchronous cyclic search. Finally, block erosion is performed, as a post-processing of the motion vector field, in order to remove spurious block structures in the interpolated frame. Using only eight candidate motion vectors per block, 3D RS successfully achieves fast convergence and reaches smooth vector field with limited complexity.

### 5.02.6.5.2 Flexible motion partitioning

Variable-size block matching, as described in [Section 5.02.6.4](#), allows for a finer motion estimation process, especially in the presence of complex scenes. Nevertheless, the subdivided structure remains based on blocks. In order to further improve motion estimation and compensation, techniques have been proposed which segment selected blocks corresponding to moving objects. Each of the segmented regions is then assigned a different motion vector.

In [55], the patterns used to partition selected blocks are derived by a vector quantization technique. A codebook is learned on segmented patterns from a training set. Similarly, in [56], blocks can be partitioned by an arbitrary line using wedges. A fast algorithm is proposed which initially detects

**FIGURE 2.22**

Template matching.

dominant edge directions in order to pre-select candidate wedges. A general framework for geometry-adaptive partitions is introduced in [57], based on a piecewise-smooth structure model.

#### 5.02.6.5.3 Template matching

In the context of video coding, most schemes estimate motion vectors at the encoder side. This data is transmitted to the decoder as **side information**, along with the **residual signal**. At the decoder side, motion vectors are then used to duplicate the prediction loop which was used at the encoder.

分类

In [58], a **decoder-side template matching** technique is introduced. The **encoder signals** for each block whether standard encoder side or proposed decoder-side mode is used. In the former case, a motion vector is estimated at the encoder and transmitted to the decoder. In the latter case, the motion vector is estimated at the decoder using template matching. More specifically, the template is defined as a causal inverse L-shaped neighborhood at the top-left of the target block (see [Figure 2.22](#)). The motion vector of the target block is then computed by minimizing a dissimilarity measure between the templates in the current and reference frames using full search.

---

#### 5.02.7 Parametric motion estimation

In this section, we consider motion estimation approaches which estimate the parameters of the motion models as defined in [Section 5.02.2.3.2](#), and more specifically in [Eqs. \(2.10\)–\(2.16\)](#). As previously discussed, these models can be applied to a [coherently moving region of support](#).

An important **special case** is when a single region of support corresponding to the whole image is selected. In this case, referred to as **global motion estimation**, the dominant motion is estimated. This dominant motion is resulting from **camera motion**, such as dolly, track, boom, pan, tilt, and roll, which is a widely used cinematic technique in filmmaking and video production.

**问题的应对** Hereafter, we describe two classes of techniques for parametric motion estimation. We also discuss difficulties arising due to **outliers**, and related **robust** estimators.

### 5.02.7.1 Indirect parametric motion estimation

A first class of approaches indirectly computes the motion parameters from a **dense motion field** rather than from the **image pixels**. More specifically, a dense motion field is first estimated, and then the parametric motion model is fitted on the obtained motion vectors.

A **Least Mean Square (LMS)** technique is commonly used for this model fitting. More specifically, the motion parameters are derived from the expressions

$$\begin{aligned} \min_{\pi} \sum_{(x,y) \in \mathfrak{N}} [u(x, y) - \hat{u}_{\pi}(x, y)]^2, \\ \min_{\pi} \sum_{(x,y) \in \mathfrak{N}} [v(x, y) - \hat{v}_{\pi}(x, y)]^2, \end{aligned} \quad (2.73)$$

where  $u(x, y)$  and  $v(x, y)$  denote the horizontal and vertical components of the dense motion field,  $\hat{u}_{\pi}(x, y)$  and  $\hat{v}_{\pi}(x, y)$  the corresponding fully parameterized motion field,  $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$  the set of parameters of the model (see [Section 5.02.2.3.2](#)), and  $\mathfrak{N}$  the support region for model fitting. The model parameters  $\pi$  can then be computed by setting to zero the partial derivatives of [Eq. \(2.73\)](#) according to  $\pi_1, \pi_2, \dots, \pi_n$ .

**步骤**

In [10, 59], the **initial** dense motion field is estimated using a gradient-based optical flow approach (see [Section 5.02.3](#)). An LMS technique is then used to compute the model parameters. The methods in [60, 61] are similar, however, a block matching technique (see [Section 5.02.6](#)) is rather used in the first step.

**缺点**

A **drawback** of these approaches is that the performance is significantly influenced by the accuracy of the **initial dense motion field**. Indeed, LMS is very sensitive to erroneous samples, which may negatively impact the model parameters estimation. Another weakness is that the region of support is assumed to be characterized by a **coherent motion** which can be closely represented by the motion model. However, this strong assumption **may not always hold**. To alleviate these two drawbacks, **robust estimation** can be used, as further discussed in [Section 5.02.7.3](#).

The same framework can also be used to estimate a parametric motion model in the **compressed domain**. In this case, block-based motion vectors are readily available from the compressed code stream. Such an approach is proposed in [62] for a low complexity global motion estimation. To take into account the high likelihood of outliers, a robust M-estimator is used. A similar compressed domain scheme is proposed in [63], relying on the Helmholtz tradeoff estimator as a robust estimator.

### 5.02.7.2 Direct parametric motion estimation

A second class of approaches directly computes the model parameters.

Based on the **optical flow equation** (see [Section 5.02.2.3.1](#)), a gradient-based formulation similar to the dense optical flow approaches is followed. However, as discussed in [Section 5.02.3](#), the optical flow equation is underconstrained. Thus, an additional constraint is required, namely a **smoothness constraint** (Horn-Schunck) or a **local uniformity constraint** (Lucas-Kanade). When including a motion model, the problem becomes implicitly constrained.

More specifically, a parametric gradient-based formulation of the optimization criterion is given by

$$\sum_{(x,y) \in \mathfrak{N}} \left[ \hat{u}_\pi(x, y) \frac{\partial f}{\partial x}(x, y) + \hat{v}_\pi(x, y) \frac{\partial f}{\partial y}(x, y) + \frac{\partial f}{\partial t}(x, y) \right]^2, \quad (2.74)$$

where  $\hat{u}_\pi(x, y)$  and  $\hat{v}_\pi(x, y)$  are the fully parameterized motion field,  $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$  the set of parameters of the model (see Section 5.02.2.3.2), and  $\mathfrak{N}$  the support region. By taking the partial derivatives of Eq. (2.74) according to the parameters  $\pi_1, \pi_2, \dots, \pi_n$ , and setting to zero, the parameters of the motion model can then be derived.

The above formulation uses the optical flow equation, which is based on a first-order Taylor series expansion (see Section 5.02.2.3.1), and is adopted in [64]. However, the first-order expansion implicitly assumes that the velocity remains small. As an alternative, a second-order Taylor series expansion is preferred in [65, 66]. In order to improve robustness, especially when estimating the gradient which is prone to noise, hierarchical schemes are used. More specifically, the process is iterated on a multi-resolution representation by means of a Gauss-Newton minimization algorithm.

A different formulation is proposed in [67, 68], where the SSD between the current frame and the motion compensated previous frame is directly minimized:

$$\sum_{(x,y) \in \mathfrak{N}} [e(x, y, t)]^2, \quad (2.75)$$

with

$$e(x, y, t) = f(x - \hat{u}_\pi(x, y), y - \hat{v}_\pi(x, y), t - 1) - f(x, y, t). \quad (2.76)$$

The motion parameters  $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$  are computed by minimizing Eq. (2.75) using the Levenberg-Marquardt iterative non-linear minimization [69]. More specifically, the following expression is obtained

$$\sum_{l=1}^n H_{k,l} \delta \pi_l = b_k, \quad (2.77)$$

where  $\delta \pi_l$  is the parameter update term

$$\delta \pi_l = \pi_l^{(i+1)} - \pi_l^{(i)}, \quad (2.78)$$

the curvature matrix  $H_{k,l}$  (equal to one-half the Hessian matrix) is defined as

$$H_{k,l} = \frac{1}{2} \sum_{(x,y) \in \mathfrak{N}} \frac{\partial^2 e^2}{\partial \pi_k \partial \pi_l} \cong \sum_{(x,y) \in \mathfrak{N}} \frac{\partial e}{\partial \pi_k} \frac{\partial e}{\partial \pi_l}, \quad (2.79)$$

and

$$b_k = \frac{1}{2} \sum_{(x,y) \in \mathfrak{N}} \frac{\partial e^2}{\partial \pi_k} = - \sum_{(x,y) \in \mathfrak{N}} e \frac{\partial e}{\partial \pi_k}. \quad (2.80)$$

Equation (2.77) is solved by Singular Value Decomposition (SVD) [69] and the process is iterated in a multi-resolution data representation. In [67], an initial matching step is included in order to find a good initial condition. Moreover, a truncated quadratic function is used in order to increase robustness to outliers.

As an alternative to the above gradient-based approaches, a **generalized matching technique** is proposed in [70]. More specifically, the model parameters are computed by minimizing a dissimilarity measure. The technique is robust, as it does not rely on a model of the luminance. However, it entails a large computational complexity.

### 5.02.7.3 Robust estimation

产生原因

**Outliers** are samples that markedly deviate from the prevailing tendency. In the case of parametric motion estimation, the presence of outliers, due to **noisy measurements** or **poorly defined support regions**, will lead to inaccurate model estimates. In the case of global motion estimation, **foreground moving objects** also correspond to outliers.

In order to alleviate the impact of outliers, robust estimation has been proposed [71, 72]. One indicator of the performance of a robust estimator is its breakdown point, roughly defined as the highest percentage of outliers that the robust estimator can tolerate.

Three classes of robust estimators can be defined:

- M-estimators: M-estimators are a **generalization of maximum likelihood estimators**. They involve the minimization of a function of the form:

$$\sum_i \rho(r_i), \quad (2.81)$$

where  $r_i$  is the **residual error** between a data sample and its fitted value, and  $\rho$  is a symmetric positive-definite function with a unique minimum at  $\rho(x = 0)$ . With a squared error function,  $\rho(x) = x^2$ , the rate of increase accelerates for large values of  $x$ , giving a very large weight to these values. Conversely, for a robust estimator, the function  $\rho$  **saturates** at large values of  $x$ .

- L-estimators: L-estimators are **linear combination** of order statistics. Two examples are the **median** and the  **$\alpha$ -trimmed-mean**.
- R-estimators: R-estimators are based on **rank tests**.

Robust estimators have been successfully used in motion estimation. Two robust estimators, the Least Median of Squares (**LMedS**) and the Least-Trimmed Squares (**LTS**), are used in [73]. Tukey's biweight function [69] is applied in an iterative re-weighting scheme in [74, 62]. A truncated quadratic function is minimized in [67]. Finally, the Helmholtz tradeoff estimator is applied in [63].

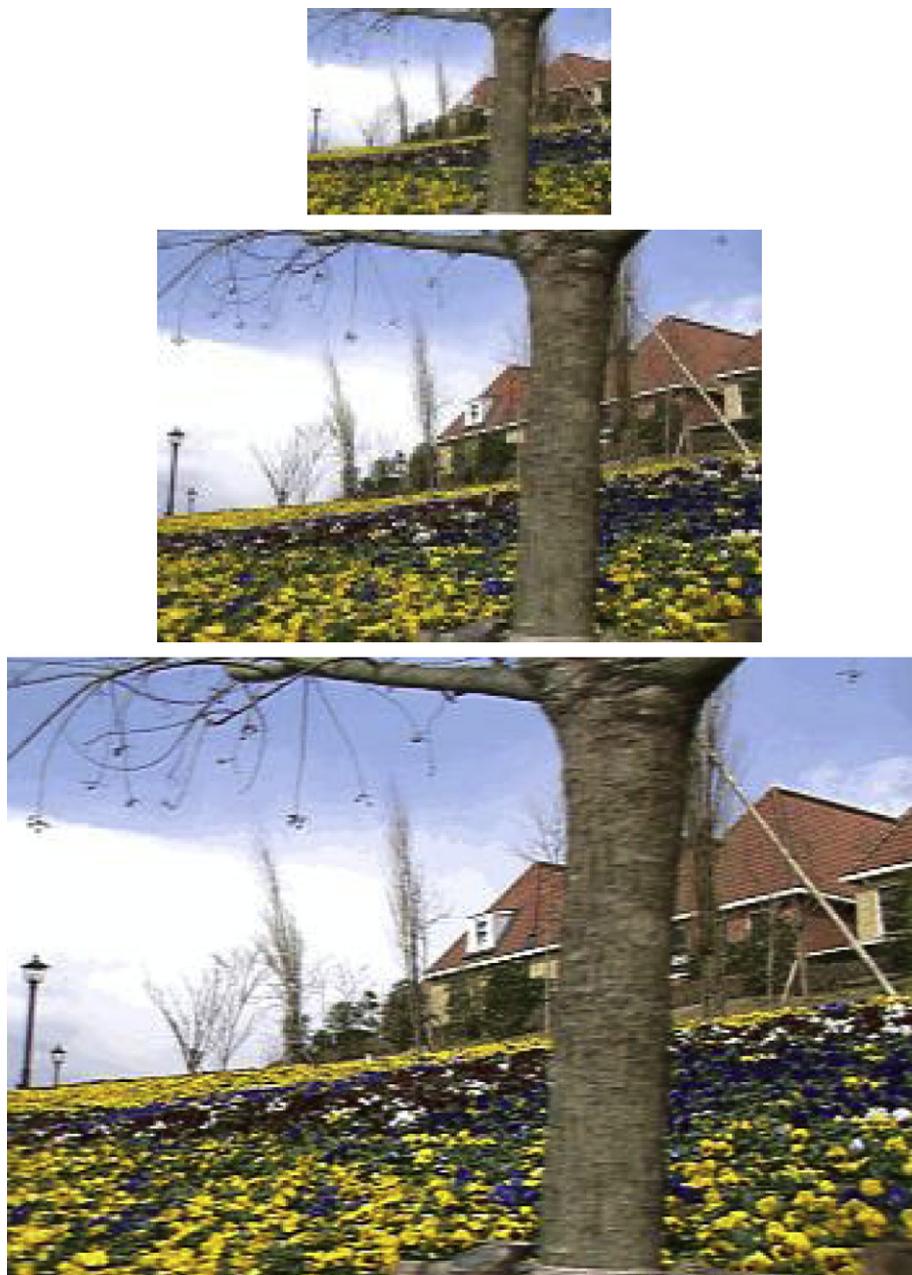
---

## 5.02.8 Multi-resolution approaches

Multi-resolution or multi-scale approaches are often used in image and video processing. They have their origin in the Laplacian pyramid introduced in [75].

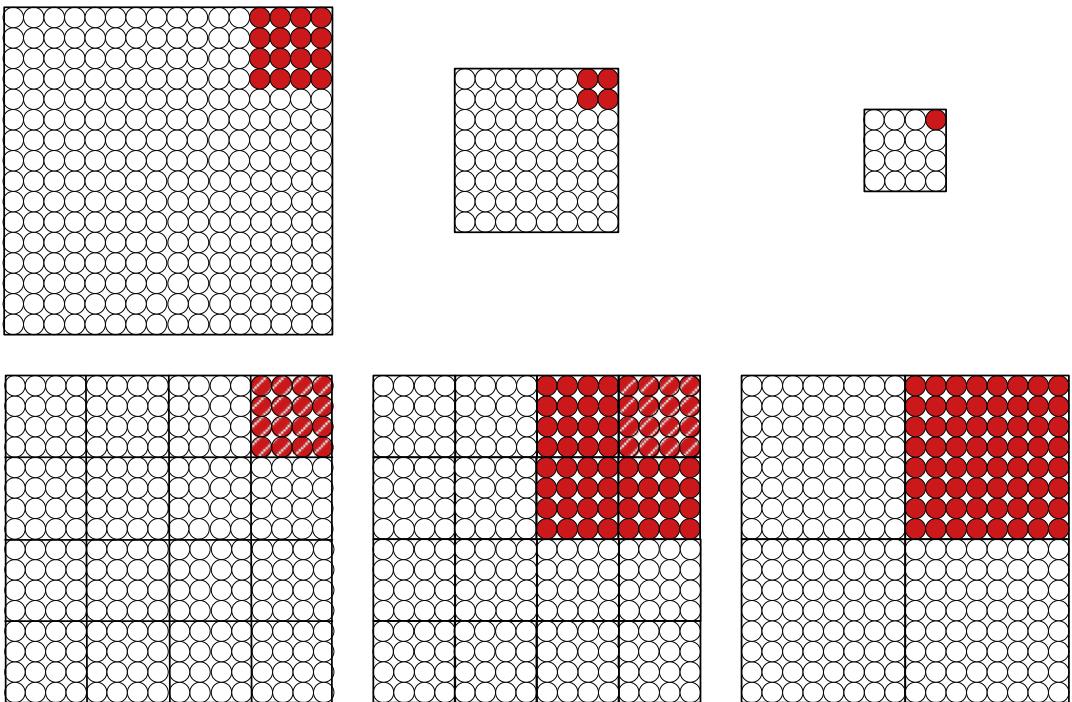
定义  
产生过程

A **low-pass pyramid** is a set of images with progressively more **smoothing** and reduced **spatial resolution**. A smoothing filter is first applied on the original image. This smoothed image is then subsampled, most commonly by a factor of two in both the horizontal and vertical directions (in this case, it is referred to as a dyadic structure). The same filtering and subsampling operations are then applied again on the resulting image, in a **recursive way**. Different smoothing kernels can be used. The concept of low-pass pyramid is illustrated in [Figure 2.23](#).



**FIGURE 2.23**

Example of low-pass pyramid (3 levels).

**FIGURE 2.24**

Duality between multiple scales (top) and single scale (bottom) for multi-resolution representation.

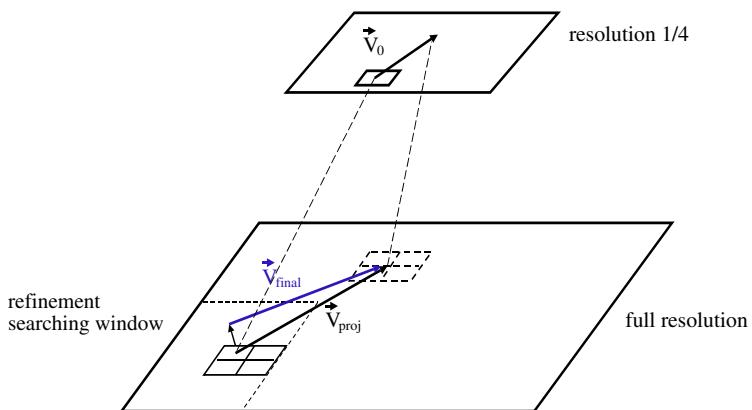
A multi-resolution and multi-scale representation is commonly used, as illustrated at the top of Figure 2.24. However, a **dual representation** at multi-resolution but a single scale is also possible, as shown at the bottom of Figure 2.24. These dual representations are **equivalent**, as the second one can be derived from the first one by upsampling and interpolation, and vice versa, the first one can be obtained from the second one by filtering and downsampling.

**优势**

In the context of motion estimation, such a multi-resolution or multi-scale representation is very appealing. Due to the **smoothing** and **spatial subsampling**, coarse resolution levels **allow efficient and robust estimation of large-scale motions**. Conversely, fine local motions can accurately be estimated at **finer resolution levels**. An additional advantage of multi-resolution motion estimation techniques is the potentially significant reduction in **computational complexity**.

Thanks to these appealing advantages, multi-resolution methods have been widely adopted for motion estimation. **Multi-resolution optical flow** methods have been proposed in [76–78]. Similarly, **multi-resolution block matching** techniques have been introduced in [3, 79, 80]. In turn, **multi-resolution parametric techniques** have been proposed in [67, 81].

Most algorithms follow a **coarse-to-fine** processing. More specifically, motion is first estimated at the coarsest resolution level. As coarse resolution input images are obtained by low-pass filtering and

**FIGURE 2.25**

Hierarchical motion estimation with refinement at high resolution.

subsampling, noise is largely smoothed out and large-range interactions can be efficiently taken into account. Hence, a robust estimation is obtained, which captures the large trends in motion. The motion field is then projected to the next finer resolution level and **iteratively refined**. This refinement allows to consider short-range relations and hence to identify **local motions** and **improve accuracy**. As a result, a more reliable motion vector field is obtained, with coherent displacements from one scale to the next.

Multi-resolution block matching motion estimation, as proposed in [79, 80], is illustrated in Figure 2.25 and described in more details. The **motivation** is to overcome the limitations of block matching techniques, and in particular to **reduce blocking artifacts** in the motion compensated frame. Block matching is first estimated on a low-pass and subsampled version of the original images, either by full-search or using some fast search techniques (see Section 5.02.6.1). Thanks to the spatial subsampling, large displacements can efficiently be estimated with a small search window. These motion vectors are then projected on the finer resolution level. If the same block size is used at both resolution levels, a parent block at the coarser level will be projected into four children blocks at the finer level. A simple projection operator is to have these four children blocks inherit the same motion vector of the parent block. Straightforwardly, its amplitude has to be multiplied by two in each direction to take into account the difference of scale. Note that it is also possible to select the best initial condition among motion vectors in a neighborhood, as discussed for variable-block size block matching in Section 5.02.6.4. The motion vectors are then refined at the finer scale. At this stage, a reduced-size search window around the current estimate can be used in order to guarantee smoothly varying motion vector fields.

While most proposed multi-resolution approaches follow the **coarse-to-fine** processing described above, some techniques have also been proposed which include **fine-to-coarse** steps. More specifically, in [77], a strategy is described which returns to coarser resolution levels when the current estimate is unreliable. In [82], a multi-grid block matching technique is described, which combines coarse-to-fine and fine-to-coarse steps in order to **avoid local minima**. Finally, an optical flow multi-grid algorithm which passes flow estimates both up and down the multi-resolution levels is proposed in [83].

It can be straightforwardly observed that a multi-resolution block matching technique, combined with a spatially adaptive grid size, is essentially equivalent to the variable-size block matching described in [Section 5.02.6.4](#).

In summary, multi-resolution or multi-scale motion estimation approaches are very appealing, as they result in more **robust** and **accurate** motion vector fields. Moreover, this performance gain is obtained with a **reduced computational complexity**. However, a **drawback** of many approaches is that the obtained motion field is overly **smooth** and sometimes fails to accurately represent **detailed structures** and **small moving objects**.

优势

伴生  
联合实现组成  
性能衡量

结果

结构，单位

过程步骤

性能，问题

解决

不确定，  
不保证

换种方式

## 5.02.9 Motion compensation

**Motion compensation** (MC) is used together with **motion estimation** (ME) to perform **temporal predictions** in the context of video compression. This approach is so effective that virtually all compression standards resort to it in order to remove the temporal redundancy from video.

**Predictive coding** consists of computing a prediction of the input signal, and in compressing the prediction error (or residual) instead of the signal itself. If the signal is sufficiently correlated in time and the prediction is computed in such a way that it could be perfectly **reproduced** at the decoder side, this predictive coding is more effective than coding the original signal.

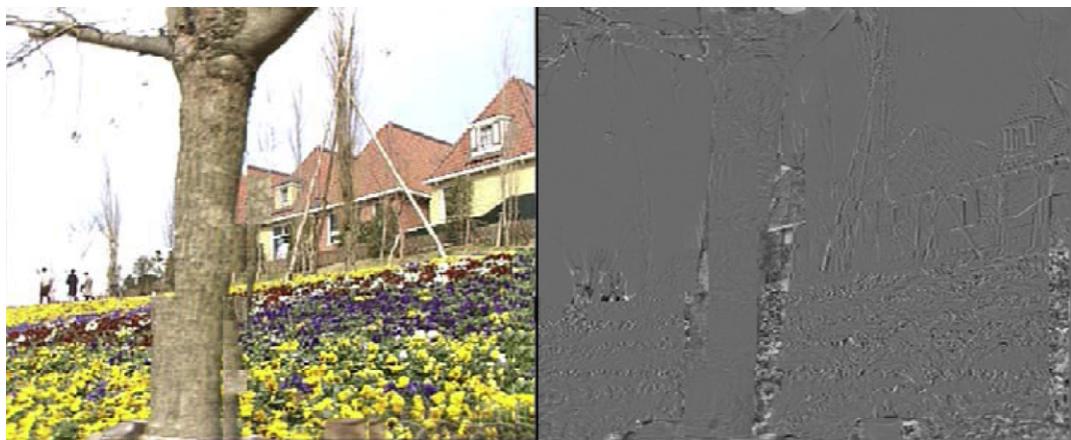
Video sequences show a very high temporal correlation, since consecutive images are very similar, and mainly differ for the movement. Therefore a motion compensated temporal prediction is generally very effective, i.e., it produces a **low-energy residual**. Frames in which blocks can be temporally predicted are called **P-frames**.

In video compression, the **coding unit** is typically a block of pixels. The current terminology for ME/MC is **macroblock**, since the term **block** was reserved for the unit of the spatial transform. However in the upcoming standard HEVC the terminology is clarified since the terms **coding unit**, **prediction unit**, and **transform unit** are explicitly introduced. We give some details about HEVC ME/MC in [Section 5.02.9.1.4](#), and until then we will keep using the terms block and macroblock.

The motion compensation consists simply in using the block from the reference image in position  $(p - \hat{i}, q - \hat{j})$  as prediction of the block of the current image in position  $(p, q)$ , where  $(\hat{i}, \hat{j})$  is the vector estimated for the position  $(p, q)$ . Therefore, instead of encoding the luminance values  $f_k(B_{p,q})$ , we have to encode  $f_k(B_{p,q}) - f_h(B_{p-\hat{i},q-\hat{j}})$ . In [Figure 2.26](#) (left) we show the motion compensated prediction of image 227 from the *flower* sequence ([Figure 2.15](#), right), produced by using the motion vector field shown in [Figure 2.16](#) (left) on the reference image (image 223, [Figure 2.15](#), left).

We observe that the prediction is good almost everywhere, except for the blocks that were **disoccluded** in the current image (i.e., appeared from behind the tree), and for those that entered the scene from the right. In order to mitigate this kind of problem, **bi-directional** motion compensated prediction has been introduced since the first video coding standards as MPEG-1: the block belonging to some images (called B-frames) can be predicted not only with blocks from a previous image, but also with blocks from a successive one. Moreover, the prediction can also consist in the average of the two blocks.

We remark that even with B-frames, **nothing assures** that a good prediction exists in the reference frames. For this reason, video standards do not oblige the encoder to use the motion compensated prediction: sometimes **it is more effective to encode the new block instead of a large prediction error**.

**FIGURE 2.26**

Motion compensated version of the reference image and the associated prediction error.

The choice between temporally predictive and non-predictive coding (also called *Intra coding*) is up to the encoder.

### 5.02.9.1 Motion compensation in H.264/AVC

**新工具**

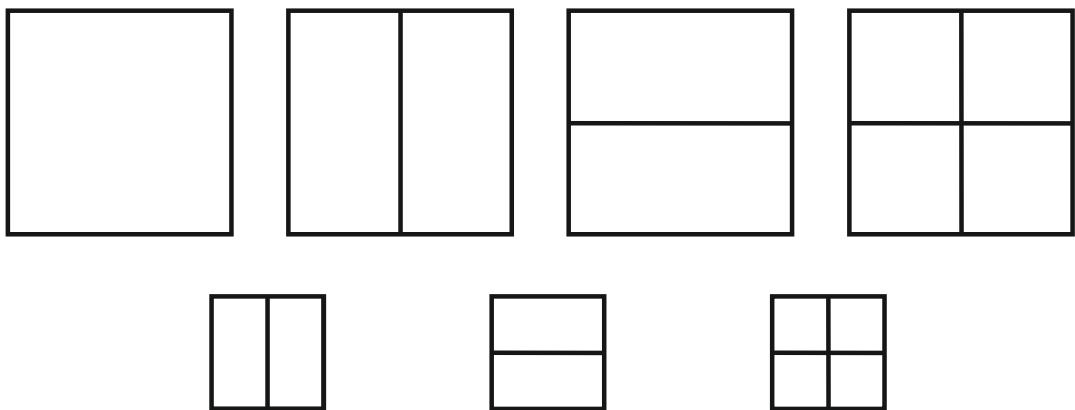
In H.264/AVC [38] the motion compensation is very effective because of the introduction of several new tools. The most relevant of these are described in the following subsections. In particular H.264/AVC uses variable block sizes for ME/MC, multiple reference frames, and quarter-pixel ME precision.

**组成**

The coding unit in H.264/AVC is the macroblock (MB), i.e., a square block of  $16 \times 16$  luminance values, plus the co-located chrominance values. For the sake of simplicity, in the following we will consider only the luminance.

**MV与大小  
关系**

A MB can be coded in several different ways, called *modes*. Motion compensation is used in temporal predictive modes, which differ in the *partition* of the MB. In the  $16 \times 16$  mode, a single motion vector is estimated for the whole MB, and the prediction is the corresponding block of pixels in the reference frame. In the other modes, the MB is divided into rectangular or square parts, and for each one a different motion vector can be encoded. Therefore the predictor of the current MB can be formed by joining blocks from different regions of the reference frame. This allows dealing successfully with blocks where several objects are present. In particular, the MB can be divided into two identical horizontal or vertical rectangles, or into four  $8 \times 8$  blocks. In this case, each of the blocks can undergo a further partition, as shown in the bottom row of Figure 2.27. In conclusion, for a single temporal predictive MB H.264/AVC allows to encode from one to sixteen motion vectors. In general one can expect that a finer partition gives a better predictor, but this comes at the cost of a higher coding rate for motion.

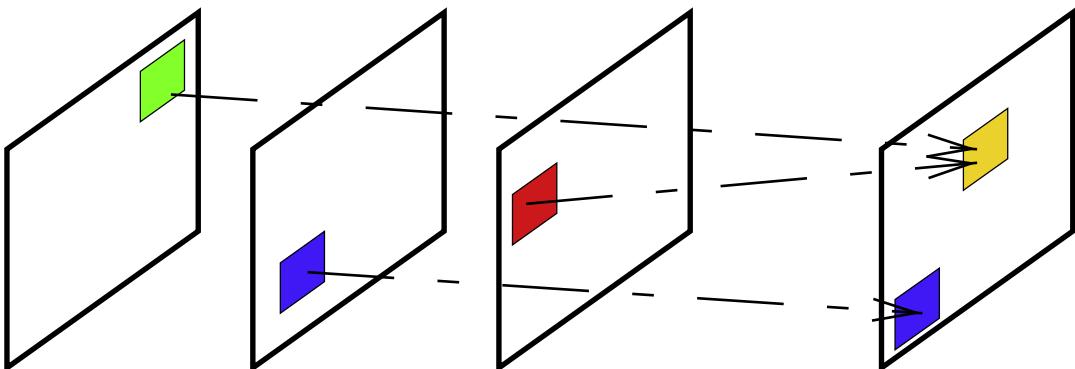
**FIGURE 2.27**

H.264/AVC macroblock partitions. First row, from left to right:  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ . If this last partition is selected, each  $8 \times 8$  block can be further partitioned into  $8 \times 4$ ,  $4 \times 8$ , or  $4 \times 4$  sub-blocks.

### 5.02.9.1.2 Multiple references and generalized P/B frames

As in previous standards, H.264/AVC allows for **mono-directional** and **bi-directional** prediction. However, in H.264/AVC the motion compensation is much more flexible than in the past.

In the case of mono-directional prediction, there is a **list** of up to 16 images: a MB can be predicted using any reference image in the list; therefore, the encoder should record in the encoded bitstream not only the **selected motion vector**, but also the **reference index** in the list. Therefore, two macroblocks in the same image can be predicted using blocks that belong to different references, as shown in **Figure 2.28**. Likewise, in the case of bi-directional prediction, **two lists** are kept. A MB can be predicted

**FIGURE 2.28**

Multiple references in H.264/AVC.

with blocks from any image in any list, or as a linear combination of a block from the first list and a block from the second one.

### 5.02.9.1.3 Rate-constraint lagrangian motion estimation

In this section we describe a non-normative tool for efficient motion estimation in H.264/AVC, referred to as rate-distortion optimization (RDO) [84].

**整体考虑**

In order to achieve the best possible RD performances, each single decision of the encoder should be taken considering the effect on the **final image quality** and **coding rate**, and then solving a constrained optimization problem: typically, the problem is the one of minimizing the distortion with a constraint on the rate. This problem can be solved with a Lagrangian approach. More precisely, given a candidate vector  $(i, j)$ , we could perform a complete encoding/decoding process, evaluating the resulting distortion and the coding rate. More precisely, we could compute the **motion compensated residual**, then perform the **spatial transform**, the **quantization**, the **inverse transform** and we could add again the prediction, obtaining the **decoded macroblock** associated to the candidate vector. The resulting **distortion** would be used as  $D(i, j)$ . At the same time, we could compute the coding rate as the sum of the rates needed to encode the **motion vector**, the **reference image index** if multiple references are possible, and the **quantized residual**. This would give  $R(i, j)$ . Finally, the best vector would be the one minimizing

$$J_{\text{RDO}}(i, j) = D(i, j) + \lambda R(i, j). \quad (2.82)$$

However, this approach is unfeasible in practice, since it would impose an extremely high **complexity**: each MB is encoded with each possible candidate vector. In practice, a good approximation of Eq. (2.82) is the following:

$$J_{\text{RDO-ME}}(i, j) = J_{\text{SAD}}(i, j) + \lambda_{\text{ME}} R_{\text{Motion}}(i, j). \quad (2.83)$$

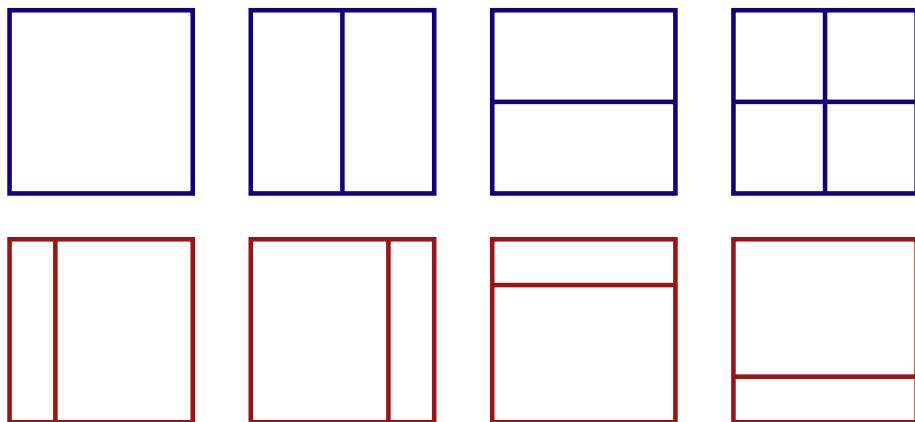
In other words we use a regularized version of the SAD criterion of Eq. (2.65), where the regularization function is the rate needed to encode the motion vector. The motion vector coding techniques for H.264/AVC and for HEVC are described in Section 5.02.9.1.5.

**决定是否  
细分**

Finally, let us say a few words about how the encoder can perform an RD-optimized selection of the **MB partition**. Since the number of partitions is relatively limited, in this case the encoder can actually use the criterion defined by Eq. (2.82). For each partition, and for each sub-block of the partition, the contributions to the total distortion and rate are computed and summed up. The mode providing the smallest total cost can be selected as the optimal encoding mode.

### 5.02.9.1.4 Preview of forthcoming HEVC

In the High Efficiency Video Coding standard [39,40], ME/MC with variable block size is further **enhanced** with respect to H.264/AVC. More precisely, **block sizes** range from  $64 \times 64$  to  $4 \times 4$ . The temporal prediction is performed within a so-called **prediction unit** that in turn can be split into one, two, or four **prediction blocks** (PBs). As in H.264/AVC, a block can be split into two rectangular blocks, but four **new splits** are possible in HEVC, in which the two rectangular blocks do not have the same number of pixels: these splits are known as **asymmetrical mode partitions**. The possible splits of a block into PBs are shown in Figure 2.29. The asymmetrical partitions have size  $M \times M/4$  or  $M \times 3M/4$ .

**FIGURE 2.29**

HEVC partition types for ME/MC. Top row: symmetrical partitions. Bottom row: asymmetrical partitions.

### 5.02.9.1.5 Motion vector coding

实现

In H.264/AVC, the motion vectors are predicted using the neighboring macroblocks' motion information available to the decoder as well. The encoder signals whether the MV prediction error is encoded (this happens in the so-called INTER modes) or not (the so-called SKIP mode in which, additionally, no transform coefficient is sent). In the first case, the prediction error is encoded using the H.264/AVC entropy coder (the simpler CAVLC or the more efficient CABAC). In the second, the prediction is used as motion vector for the current block.

变量

The predictor changes according to the MB partition and to the availability of MVs in adjacent MBs. For example, for rectangular partitions, the predictor is the MV of a single neighboring MB. For square partitions, up to three neighboring vectors can be collected, and their median is used as motion vector predictor.

几种模式

In HEVC, motion information can be coded using one out of two different modes. In the Merge mode, the encoder builds a list of candidates that can be exactly reproduced by the decoder as well; then, the encoder only needs to send an index to single out the selected MV in the list. Note that, instead of producing a single prediction and then correcting it, several “predictions” are produced, but it is not possible to modify them. If the candidates are well chosen, one (or more) of them could efficiently represent the motion of the current block. In order to exploit the uniformity of motion, the candidates are chosen among the spatial neighbors of the current block, the co-located temporal neighbor, and a set of predefined vectors.

In HEVC, the SKIP mode can be seen as a special case of the Merge mode; moreover, a Nonmerge mode exists which is based on MV prediction and differential coding, as in H.264/AVC. The predictor is one out of two possible candidates that in turn are selected among the spatial candidates of the Merge mode. A temporal candidate is included if there are not enough different spatial candidates.

### 5.02.9.2 Overlapped Block Motion Compensation

When one applies motion compensation on an image using a motion vector field produced with block matching, the resulting image can be affected by **blocking artifacts**. Motion compensation consists in copying blocks from disparate locations in the reference image and in putting them side-by-side: of course, nothing assures a smooth transition between them. Therefore **unnatural image luminance variations** appear in correspondence of the **block grid**, giving rise to annoying visual artifacts, as shown for example in [Figure 2.26](#) (left).

In order to overcome the block artifacts in the motion compensated frame, **Overlapped Block Motion Compensation (OBMC)** has been proposed [85]. This method simply consists in considering for the computation of the motion prediction at the boundaries of a block not only the contribution of the estimated block, but also that of **neighboring blocks**, leading to a prediction by a **weighted average** of these two contributions. A schematic example is shown in [Figure 2.30](#).

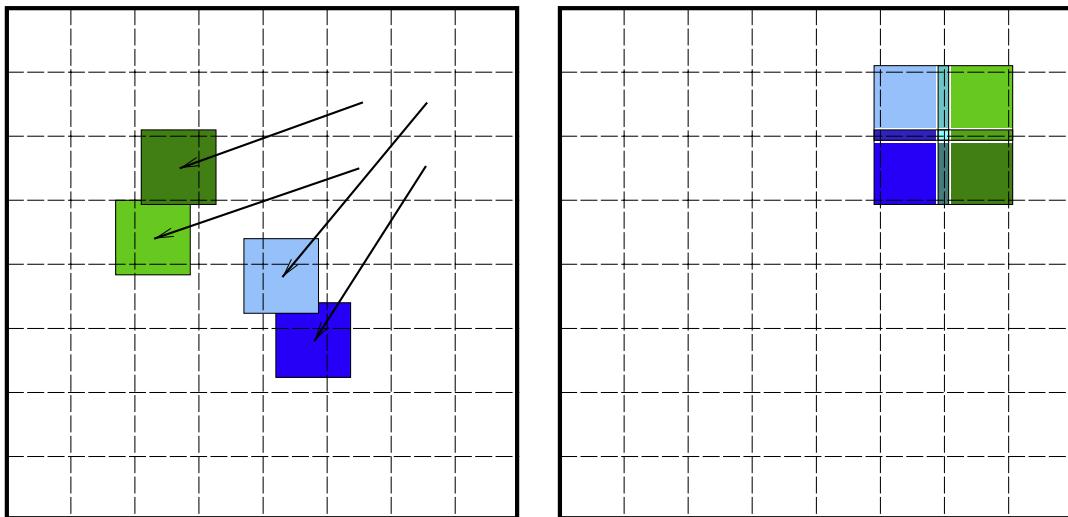
An estimation-theoretic analysis of motion compensation is presented in [86]. OBMC is formulated as a **probabilistic linear estimator of pixel intensities**, which leads to improved prediction.

### 5.02.9.3 Global Motion Compensation

**Global Motion Compensation (GMC)** is especially suited to encode video content with a significant amount of **camera motion**, such as panning, zooming, and tilting. In such a case, the coding efficiency of common **Local Motion Compensation (LMC)** decreases. On the one hand, a large number of motion

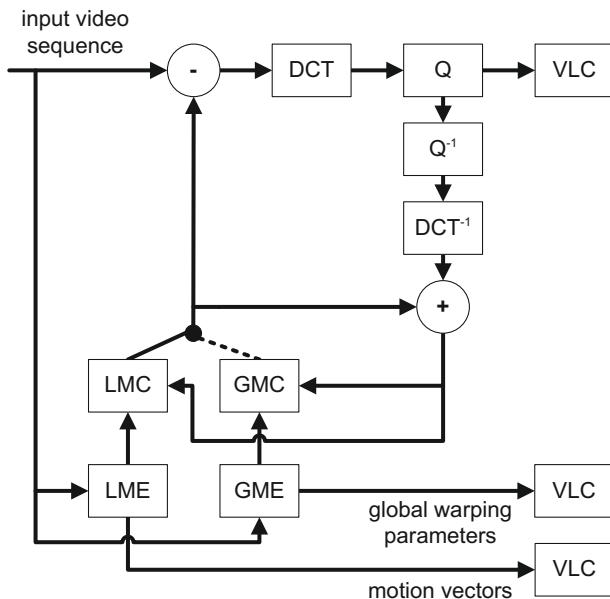
结果  
产生原因  
产生位置

镜头变换  
导致结果  
变差原因



**FIGURE 2.30**

Principle of Overlapped Block Motion Compensation. Left: We show four motion vectors and the blocks they point toward in the reference image. Right: After motion compensation, the blocks partially overlap, allowing a smooth transition from one block to another.

**FIGURE 2.31**

MPEG-4 GMC encoder.

实现

vectors have to be transmitted for the moving background. On the other hand, a translational motion model may fail in the presence of camera motion including zoom or rotation. GMC improves upon LMC by building a prediction using **global motion parameters**. In this way, the **dominant motion** is accurately represented with very few parameters.

GMC has been standardized in MPEG-4 [87], more precisely in the **Advanced Simple Profile**. We now describe this scheme in more detail. The encoder is illustrated in Figure 2.31. More precisely, it involves the following steps. **Both** Local Motion Estimation (LME) (i.e., baseline block matching, see Section 5.02.6) and Global Motion Estimation (GME) (see Section 5.02.7) are **performed**. Next, the encoder **selects** for each MacroBlock (MB) the best prediction between LMC and GMC. **Global** motion parameters are encoded and transmitted for **every frame**. In addition, motion vectors are transmitted for every MB encoded using LMC.

选择

The LMC—GMC decision is not normative. In the MPEG-4 Verification Model [88], the following test is used

$$\text{if } (\text{SAD}_{\text{GMC}} - P(Q_p, \text{MV}_x, \text{MV}_y)) < \text{SAD}_{\text{LMC}} \text{ use GMC,} \\ \text{otherwise use LMC,} \quad (2.84)$$

where  $\text{SAD}_{\text{GMC}}$  (respectively  $\text{SAD}_{\text{LMC}}$ ) is the sum of absolute difference between the original MB and the GMC prediction (respectively the LMC prediction),  $Q_p$  is the quantization parameter, and  $(\text{MV}_x, \text{MV}_y)$  is the motion vector obtained by LME. The term  $P(Q_p, \text{MV}_x, \text{MV}_y)$  is defined as

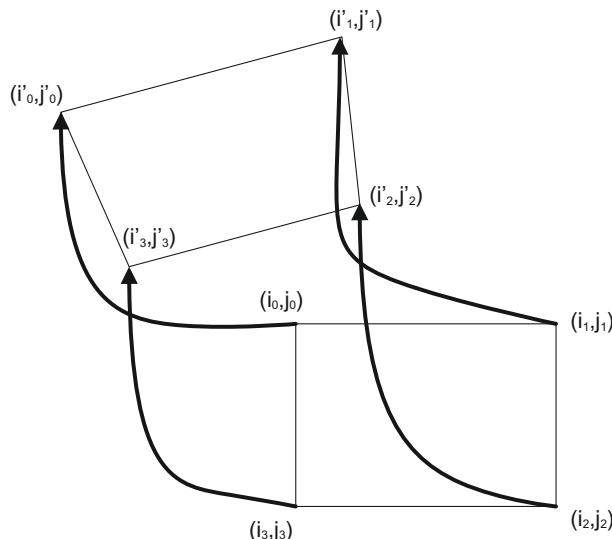
$$P(Q_p, \text{MV}_x, \text{MV}_y) = (1 - \delta(\text{MV}_x, \text{MV}_y))(N_B Q_p)/64 \\ + 2\delta(\text{MV}_x, \text{MV}_y)(N_B/2 + 1), \quad (2.85)$$

目的  
设计原理编码对象  
传输对象

with  $N_B$  the number of non-transparent pixels in the macroblock (MPEG-4 supports object-based video coding with arbitrary-shape objects), and  $\delta(MV_x, MV_y) = 1$  when  $(MV_x, MV_y) = (0, 0)$  and 0 otherwise. The purpose of this term is to give an edge to GMC, especially when  $O_B$  is large (i.e., at low bit rate). It is motivated by the two following observations. Firstly, the gain brought by GMC is in large part due to a reduced amount of overhead motion information since no motion vector is transmitted for MB encoded using the GMC mode. Secondly, the gain resulting from GMC increases at very low bit rates, as in this case the bit rate to transmit motion vectors becomes a larger percentage of the overall bit rate.

The global motion parameters have to be transmitted to the decoder. For this purpose, instead of directly transmitting the parameters of the motion model, displacement of  $n$  reference points is encoded, with  $n = 1, \dots, N$ . More precisely, reference points  $(i_n, j_n)$  are positioned at the corners of the current frame (or the bounding box in the case of an arbitrary-shape object), and the corresponding points  $(i'_n, j'_n)$  are computed in the reference frame using the global motion parameters, as shown in Figure 2.32. Next, the coordinates  $(i'_n, j'_n)$  are quantized to half-pel precision. Finally, the vectors  $(u_n, v_n) = (i_n - i'_n, j_n - j'_n)$  are computed and transmitted as differential motion vectors. Four motion models are considered: perspective model where  $N = 4$  pixels are enough to estimate the model parameters, affine ( $N = 3$ ), translation—isotropic magnification—rotation ( $N = 2$ ), and translation ( $N = 1$ ).

Similar to the GMC in MPEG-4, a two-stage motion compensation for H.263 was introduced in [89]. First, GMC is applied to the reference frame, giving a new GMC reference. Next, LMC is performed twice: on the GMC reference and on the regular reference. The best prediction is then selected.

**FIGURE 2.32**

Trajectories encoding.

The scheme in [90] combines affine MC with long-term memory MC prediction. Several sets of affine parameters are estimated for sub-regions of the image. Then, for each affine set, the reference picture is warped and added in the long-term memory buffer. Conventional block-based ME and MC is then carried out using all the available reference pictures, and RDO is used for optimal mode selection. In [91], a new prediction mode is proposed which combines GMC and temporal filtering of the previously decoded pictures. A rate-distortion optimization is applied on each macroblock to decide whether to use this new prediction mode. The technique is integrated in H.264/AVC, showing improved coding performance.

Exploiting global motion information, an adaptive temporal filter is proposed in [92], as a post-processing for HEVC.

One of the major drawbacks of GMC is an increased computational complexity both at the encoder and decoder sides. The gain achieved straightforwardly depends on the type of motion in the sequence.

#### 5.02.9.4 Sprites

A sprite, also known as mosaic or panoramic image, refers to a large composite image obtained by aligning and blending pixels from different video frames, see [93–95]. In the presence of significant camera motion, a sprite can often reconstruct a large panoramic view of the background of the scene by estimating global motion (see Section 5.02.7). This sprite efficiently captures temporal information, resulting in a very compact representation.

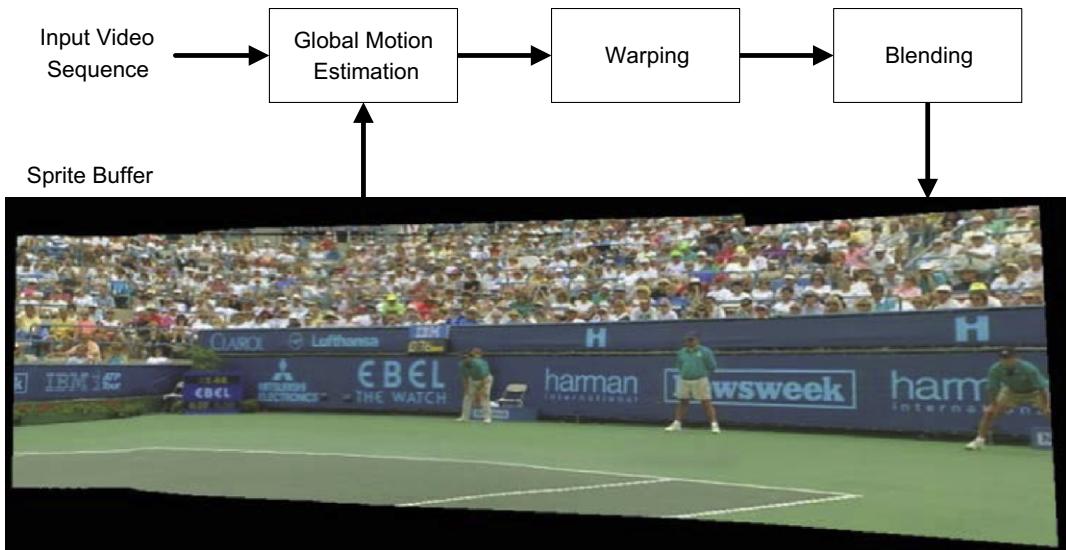
As an alternative to GMC, but with a similar objective, sprite coding has been proposed as an efficient way to represent a video sequence [96–98]. Sprite coding has been standardized in MPEG-4 [87].

The sprite has typically to be generated off-line, prior to sprite coding. The process is illustrated in Figure 2.33. Global motion estimation is first performed. Techniques such as those presented in Section 5.02.7 can be used, usually with a translational, affine, or perspective model. Warping [99] is then used to align pixels of the current video frame with the background sprite. Finally, the aligned frames are blended and accumulated in the sprite. Note that the sprite is constructed for a region characterized by a coherent motion. This may require a segmentation step, for instance to identify background and foreground regions.

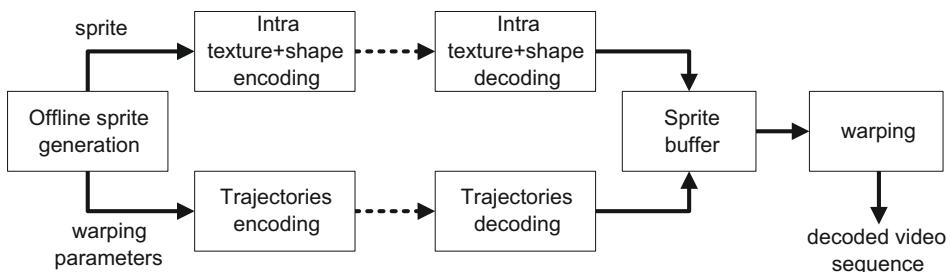
本质，描述

The sprite is in fact a large still image with an associated segmentation mask. In MPEG-4 terminology, it is referred to as a Video Object Plan (VOP) [87]. Its texture and shape can efficiently be encoded using an Intra coding technique. It is then sufficient to transmit the sprite, along with the warping parameters, resulting in a very compact representation. At the decoder side, frames of the video sequence can be reconstructed using the warping parameters and the sprite content. The MPEG-4 sprite encoding and decoding is illustrated in Figure 2.34 [87]. Trajectories are encoded in the same way as for the GMC technique in MPEG-4 (see Section 5.02.9.3).

In [96], the sprite is dynamically built and used in a motion compensated predictive scheme. In order to improve the sprite generation process, a more efficient blending technique is introduced in [100], based on the region reliability. Furthermore, an arbitrary-shape spatial prediction method is proposed to increase the coding performance. A scheme combining sprite coding with automatic background subtraction and H.264/AVC is presented in [101]. A rate-distortion optimization technique is also introduced. A long-term global motion estimation technique is proposed in [102], using a closed-loop

**FIGURE 2.33**

Sprite generation process.

**FIGURE 2.34**

Sprite coding in MPEG-4.

prediction to avoid accumulation of errors, which is especially fitted for sprite coding. An overview of sprite generation and coding is presented in [103], along with some recent developments.

场景

Sprite coding can efficiently encode **synthetic graphic objects**. In the case of natural video, the sprite has to be constructed **off-line** prior to coding. Therefore, it makes the approach **unsuitable** for real-time applications. Moreover, sprite coding is only fitting for a video object whose motion can be approximated by a rigid 2-D model. However, this assumption very often holds true for the background.

## 5.02.10 Performance assessment criteria for motion estimation algorithms

It is quite difficult to assess the performance of a motion estimation algorithm without including it in a specific application. As we have previously discussed, different properties are desirable when considering image sequence analysis or video coding.

In the context of image sequence analysis, the ability to provide a very accurate motion vector field is primordial, even though the resulting field is dense and more costly to encode. Tests can be set up, implementing artificial motions, in order to check the performance of the algorithm by comparing true and estimated motion vector fields.

When considering video coding, the primary objective is to achieve optimal rate-distortion coding performance. In this case, the ability to reliably estimate the motion present in the scene remains a secondary goal.

首要目的  
次要目的

### 5.02.10.1 Assessment of optical flow techniques

A quantitative assessment is introduced in [19], along with a comparative analysis of several optical flow motion estimation algorithms. However, a very limited data set is used.

More recently, an evaluation methodology is introduced in [20]. The first step is to collect a ground-truth data set. A key difficulty is to be able to derive ground-truth for the optical flow. In order to cover a broad range of content with varying characteristics, four types of video data are considered:

- Real imagery of non-rigidly moving scene: Test video sequences are captured in visible light. In parallel, a dense optical flow ground-truth is captured in UV light using hidden fluorescent painted texture.
- Realistic synthetic imagery: Synthetic sequences are useful, as the motion can be precisely determined. Synthetic sequences obtained by rendering complex scenes with varying amount of motion, realistic textures, and occlusions are considered.
- Imagery for frame interpolation: Test sequences are temporally decimated. The temporally up-converted sequences obtained by frame interpolation can then be compared to the corresponding original sequences. In other words, instead of directly comparing the precision of the obtained motion vectors with a ground-truth field, it is proposed to evaluate the ability of the optical flow to provide an accurate motion compensated interpolated frame, which may be more important in many application scenarios [104].
- Real stereo imagery of rigid scenes: Dense disparity ground-truth is captured for pairs of stereo images, using structured light [105].

This test data set is publicly available at <http://vision.middlebury.edu/flow/>.

We now discuss the methodology and measures used to assess performance. The Angular Error (AE) is often used to compare estimated and ground-truth flows. Let us define  $(u_{GT}, v_{GT})$  the ground-truth

optical flow, and  $(u, v)$  the estimated one. AE is then defined as

$$\text{AE} = \cos^{-1} \left( \frac{1 + u \cdot u_{\text{GT}} + v \cdot v_{\text{GT}}}{\sqrt{1 + u^2 + v^2} \sqrt{1 + u_{\text{GT}}^2 + v_{\text{GT}}^2}} \right). \quad (2.86)$$

The denominator has for purpose to **avoid** a divide by **zero** in case of a null motion vector. However, it has for consequence to arbitrarily weight differently errors depending on the amplitude of the motion vector.

Avoiding this shortcoming, and thus probably more appropriate, the **Error in flow Endpoint** (EE) is defined as

$$\text{EE} = \sqrt{(u - u_{\text{GT}})^2 + (v - v_{\text{GT}})^2}. \quad (2.87)$$

When considering the frame interpolation scenario, the **Interpolation Error** (IE) is defined as the RMS difference between the ground-truth image,  $f_{\text{GT}}(x, y)$ , and the motion compensated interpolated one  $\hat{f}(x, y)$ ,

$$\text{IE} = \sqrt{\frac{1}{N} \sum (\hat{f}(x, y) - f_{\text{GT}}(x, y))^2}. \quad (2.88)$$

Alternatively, a gradient-normalized RMS is used in the **Normalized interpolation Error** (NE),

$$\text{NE} = \sqrt{\frac{1}{N} \sum \frac{(\hat{f}(x, y) - f_{\text{GT}}(x, y))^2}{\|\nabla f_{\text{GT}}(x, y)\|^2 + \epsilon}}. \quad (2.89)$$

To take into account the observation that optical flow estimation is harder in some regions of the image, error measure statistics are computed over three **types** of regions: **around motion discontinuities**, in **texture-less regions**, and **over the whole image**.

This evaluation methodology has been widely used by other researchers in the field, who frequently use the publicly available data set to report the performance of their algorithm. Many researchers have also uploaded their results to the Middlebury website.

### 5.02.10.2 Assessment of motion estimation for video coding

In the context of video coding, the quality of the estimated motion field can be evaluated, similarly to [Section 5.02.10.1](#), in order to assess the ability of the method to estimate the true motion in the scene. In particular, **smooth motion vector fields** are desired in order to prevent artificial discontinuities in the **DFD** and to reduce the overhead needed to transmit the motion information. A second measure of the quality of the motion estimation algorithm is the **energy** of the DFD, giving insight about the quality of the prediction.

However, a more relevant criterion to evaluate a motion estimation algorithm is to consider the **global rate-distortion performance** when it is used in a given video coding scheme. More specifically, the following aspects have to be considered:

- **Objective quality metric:** In order to assess the rate-distortion performance of a video coding scheme, the distortion, or equivalently the quality, of the reconstructed sequence should be estimated. **PSNR** is the most commonly used objective quality metric. It is computed between the reconstructed and

original sequences, frame by frame, and in a range of bit rates of interest. However, it has been well documented that PSNR is not always well correlated with the perceived visual quality [106]. It is largely due to the fact that PSNR totally ignores properties of the HVS.

For this reason, perceptually driven objective quality metrics have been proposed, for instance **SSIM** [107], **VIF** [108], or **PSNR-HVS-M** [109]. These metrics typically achieve better correlation with perceived visual quality when compared to PSNR.

- Subjective quality assessment: Given the limitations of objective quality metrics, subjective tests are needed in order to reliably and thoroughly assess the visual quality of a video sequence. For this purpose, several **protocols** have been defined, for instance for TV applications (ITU-R BT.500-12 [110]) or multimedia applications (ITU-R P.910 [111]).

In the context of motion estimation, it is particularly essential to visually inspect the reconstructed video sequence for the presence of distortions resulting from failures of the motion estimation technique.

矛盾

Moreover, a **mismatch** between the motion estimation technique and the coding strategy is another potential cause of distortions. For instance, visible blocking artifacts may be introduced in the case of a block-based motion estimation which, when followed by a wavelet-based coding of the residual signal, i.e., a transform involving the entire image, may have a worse effect than in the case of an hybrid coding scheme (see also [Section 5.02.6.2.4](#)).

Note that video coding standards, such as H.264/AVC or HEVC, only specify the minimum requirements to guarantee interoperability. In other words, only the **syntax** and **semantic** of the **code stream**, along with the decoding process, are **normative**. Therefore, although the specific motion estimation technique used may significantly influence the performance of a video encoder, this part of the encoding process is not in the normative scope of the standard.

## 5.02.11 Summary and concluding remarks

In this chapter, we have reviewed some of the most important techniques for motion estimation. Motion estimation plays an important role in a broad range of applications encompassing image sequence analysis, computer vision, and video communication. As these domains entail different requirements and constraints in terms of performance, we have mainly taken here a video coding viewpoint.

As a **preamble**, we have first discussed the **notion** of apparent motion or optical flow. We have also introduced different models for motion representation.

After these **preliminaries**, we have discussed the main **approaches** for motion estimation: gradient-based techniques solving the optical flow equation, pel-recursive techniques iteratively minimizing the DFD, transform-domain techniques applied on Fourier, DCT, or DWT coefficients, block matching techniques which are widely adopted in video coding schemes, and finally parametric techniques to estimate the parameters of a motion model.

As a key component in image and video processing, we then described the concept of multi-resolution or multi-scale approaches, which typically leads to more accurate and robust motion vector fields, along with reduced computation complexity.

Next, given our emphasis on video coding applications, we explained in more details different methods for motion compensation, including a more thorough description of the various modes enabled in the state-of-the-art H.264/AVC and HEVC video coding standards.

To complete the chapter, we finally discussed methodologies to effectively assess the performance of motion estimation algorithms.

Motion estimation is a complex subject. In this chapter, we have aimed at a broad and comprehensive overview, although it is certainly not exhaustive. We have more thoroughly discussed some of the fundamental algorithms, and we have complemented them with descriptions of more recent and advanced developments. The large number of references gives the reader the opportunity to further explore different aspects and directions.

With continuous improvements over the last decades, current state-of-the-art motion estimation algorithms typically achieve good performances. Nevertheless, a general standard motion estimation technique remains elusive. In particular, diverse applications imply very different requirements and properties. However, further enhancements can be expected in the future.

## Relevant websites

Optical Flow—Middlebury Computer Vision: <http://vision.middlebury.edu/flow/>

Optical Flow Algorithm Evaluation: <http://of-eval.sourceforge.net/>

The Moving Picture Experts Group website: <http://mpeg.chiariglione.org/>

## Glossary

<b>2D motion vector field</b>	the 2D motion vector field is defined as the projection of the 3D objects motion onto the 2D image plane
<b>Aperture problem</b>	when observing a moving structure through a small aperture, different physical motions appear indistinguishable
<b>Block matching</b>	motion estimation algorithms based on the matching of blocks between two frames, with the objective to minimize a dissimilarity measure
<b>Brightness constancy</b>	assumption that a pixel intensity remains constant along a motion trajectory. In other words, variations in time of the pixel intensity are exclusively due to the objects displacements
<b>Dense motion field</b>	motion field which represents motion by assigning one motion vector to each image pixel
<b>Lambertian reflectance</b>	an ideal diffusely reflecting surface, whose apparent brightness is the same, regardless of an observer view angle
<b>Motion compensation</b>	to perform temporal processing on pixels along motion trajectories, i.e., shifted by a motion vector, instead of on co-located pixels. In video coding, motion compensated prediction refers to the prediction of a block by using a shifted previously encoded block, where the shift corresponds to the estimated motion vector
<b>Motion trajectory</b>	the path that a pixel follows through space and time when considering an image sequence as a three-dimensional continuous spatio-temporal field

<b>Multi-resolution motion estimation</b>	techniques based on a multi-resolution or multi-scale data representation, which first compute a coarse estimate of the motion field at the lowest resolution level and then progressively refine it at successively higher resolution levels
<b>Occlusion, Disocclusion</b>	an occlusion refers to a region or object which is partially or fully hidden by another object closer to the camera. A disocclusion denotes a newly appearing region or object which was previously occluded
<b>Optical flow</b>	defined as the apparent motion of the brightness pattern. In other words, the optical flow captures the spatio-temporal variation of pixel intensities in an image sequence
<b>Outlier</b>	sample which markedly deviates from the rest of the data samples
<b>Sprite</b>	a sprite, also known as mosaic or panoramic image, refers to a large composite image obtained by aligning and blending pixels from multiple displaced images
<b>Rate-distortion</b>	in the context of lossy data compression, the optimal minimization of the date rate in order to be able to reconstruct the source without exceeding a given distortion, or reciprocally, the minimization of the reconstruction distortion for a given data rate
<b>Region of support</b>	the region of support is the set of image pixels to which a motion model applies
<b>Parametric motion model</b>	represents the motion of a region characterized by a coherent motion with a set of parameters, also known as motion parameters

---

## References

- [1] E.C. Hildreth, The Measurement of Visual Motion, The MIT Press, Cambridge, MA, 1984.
- [2] S. Ullman, The Interpretation of Visual Motion, The MIT Press, Cambridge, MA, 1979.
- [3] F. Dufaux, F. Moscheni, Motion estimation techniques for digital TV: a review and a new contribution, Proc. IEEE 83 (6) (1995) 858–876.
- [4] D.J. Fleet, Y. Weiss, Optical flow estimation, in: Y. Paragios et al. (Eds.), Handbook of Mathematical Models in Computer Vision, Springer, 2006, ISBN 0-387-26371-3.
- [5] C. Stiller, J. Konrad, Estimating motion in image sequences: a tutorial on modeling and computation of 2D motion, IEEE Signal Process. Mag. 16 (4) (1999) 70–91.
- [6] G. Tziritas, C. Labit, Motion Analysis for Image Sequence Coding, Advances in Image Communication, Elsevier, 1994.
- [7] B.K.P. Horn, Robot Vision, The MIT Press, Cambridge, 1986, pp. 278–298.
- [8] B.K.P. Horn, B.G. Schunck, Determining optical flow, Artif. Intell. 17 (1981) 185–203.
- [9] N. Diehl, Object-oriented motion estimation and segmentation in image sequences, Signal Process. Image Commun. 3 (1) (1991) 23–56.
- [10] J.Y.A. Wang, E.H. Adelson, Representing moving images with layers, IEEE Trans. Image Process. 3 (5) (1994) 625–638.
- [11] F. Dufaux, F. Moscheni, Segmentation-based motion estimation for second generation video coding techniques, in: L. Torres, M. Kunt (Eds.), Second Generation Video Coding Techniques, Kluwer Academic Publishers, 1996, pp. 219–263.

- [12] M.M. Chang, A.M. Tekalp, M.I. Sezan, Simultaneous motion estimation and segmentation, *IEEE Trans. Image Process.* 6 (9) (1997) 1326–1333.
- [13] F. Zhang, D.R. Bull, A parametric framework for video compression using region-based texture models, *IEEE J. Sel. Top. Signal Process.* 5 (7) (2011) 1378–1392.
- [14] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: *Proceedings of Imaging Understanding Workshop*, 1981, pp. 121–130.
- [15] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping , in: *Proceedings of the European Conference Computer Vision*, 2004.
- [16] S. Seitz, S. Baker, Filter flow, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2009.
- [17] H. Zimmer, A. Bruhn, J. Weickert, B.R.L. Valgaerts, A. Salgado, H.-P. Seidel, Complementary optic flow, in: *Proceedings of the International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2009.
- [18] L. Xu, J. Jia, Y. Matsushita, Motion detail preserving optical flow estimation, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (9) (2012) 1744–1757.
- [19] J.L. Barron, D.J. Fleet, S. Beauchemin, Performance of optical flow techniques, *Int. J. Comp. Vis.* 12 (1) (1994) 43–77.
- [20] S. Baker, D. Scharstein, J.P. Lewis, et al., A database and evaluation methodology for optical flow, *Int. J. Comput. Vis.* 92 (2011) 1–31.
- [21] A. Netravali, J.D. Robbins, Motion-compensated television coding part I, *Bell Syst. Tech. J.* 58 (3) (1979) 629–668.
- [22] C. Cafforio, F. Rocca, The differential method for motion estimation, in: T.S. Huang (Ed.), *Image Sequence Processing and Dynamic Scene Analysis*, Springer Verlag, New York, 1983, pp. 104–124.
- [23] D.R. Walker, K.R. Rao, Improved Pel-recursive motion-estimation, *IEEE Trans. Commun.* 32 (10) (1984) 1128–1134.
- [24] J. Biemond, L. Looijenga, D.E. Boekee, A pel-recursive Wiener-based displacement estimation algorithm, *Signal Process.* 13 (4) (1987) 399–412.
- [25] S.N. Efstratiadis, A.K. Katsaggelos, A model-based pel-recursive motion estimation algorithm, in: *IEEE Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, 1990.
- [26] S.N. Efstratiadis, A.K. Katsaggelos, An adaptive regularized recursive displacement estimation algorithm, *IEEE Trans. Image Process.* 2 (1993) 341–352.
- [27] D.J. Fleet, A.D. Jepson, Computation of component image velocity from local phase information, *Int. J. Comput. Vis.* 5 (1990) 77–104.
- [28] B.G. Haskell, Frame-to-frame coding of television pictures using two-dimensional Fourier transforms, *IEEE Trans. Inf. Theory* 20 (1) (1974) 119–120.
- [29] U.V. Koc, K.J. Ray Liu, Discrete-Cosine/Sine transform based motion estimation, in: *IEEE Proceedings of the International Conference on Image Processing*, Austin, TX, November 1994, pp. 771–775.
- [30] I. Daubechies, The wavelet transform, time-frequency localization and signal analysis, *IEEE Trans. Inf. Theory* 36 (5) (1990) 961–1005.
- [31] J. Skowronski, Pel recursive motion estimation and compensation in subbands, *EURASIP Signal Process.* 14 (1999) 389–396.
- [32] A. Fuldsæth, T.A. Ramstad, Subband video coding with smooth motion compensation, in: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Atlanta, USA, May 1996.
- [33] C. Bernard, Wavelets and ill posed problems: optic flow and scattered data interpolation (Ph.D. thesis), Ecole Polytechnique, 1998.

- [34] J. Magarey, N. Kingsbury, Motion estimation using a complex-valued wavelet transform, *IEEE Trans. Signal Process.* 46 (4) (1998) 1069–1084.
- [35] E.P. Simoncelli, Bayesian multiscale differential optical flow, in: H. Jähne, Geissler (Eds.), *Handbook of Computer Vision and Applications*, Academic Press, 1998.
- [36] J. Weber, J. Malik, Robust computation of optical flow in a multi-scale differential framework, *Int. J. Comput. Vis.* 14 (1) (1995) 5–19.
- [37] C. Bernard, Fast optic flow computation with discrete wavelets Technical Report, CMAP, Ecole Polytechnique, 1997.
- [38] T. Wiegand, G.J. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 560–576.
- [39] J.-R. Ohm, G.J. Sullivan, High efficiency video coding: the next frontier in video compression, *IEEE Signal Process. Mag.* 30 (1) (2013) 152–158.
- [40] G.J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Trans. Circ. Syst. Video Technol.* 22 (12) (2012) 1649–1668.
- [41] J.R. Jain, A.K. Jain, Displacement measurement and its application in interframe image coding, *IEEE Trans. Commun.* 29 (12) (1981) 1799–1808.
- [42] M. Ghanbari, The cross-search algorithm for motion estimation, *IEEE Trans. Commun.* 38 (1990) 950–953.
- [43] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, T. Ishiguro, Motion compensated interframe coding for video conferencing, in: Proceedings of National Telecommunication Conference, New Orleans, LA, November–December 1981, pp. G5.3.1–G5.3.5.
- [44] R. Li, B. Zeng, M.L. Liou, A new three-step search algorithm for block motion estimation, *IEEE Trans. Circ. Syst. Video Technol.* 4 (1994) 438–442.
- [45] L.M. Po, W.C. Ma, A novel four-step search algorithm for fast block motion estimation, *IEEE Trans. Circ. Syst. Video Technol.* 6 (1996) 313–317.
- [46] S. Zhu, K.-K. Ma, A new diamond search algorithm for fast block-matching motion estimation, *IEEE Trans. Image Process.* 9 (2) (2000) 287–290.
- [47] C. Zhu, X. Lin, L.-P. Chau, Hexagon-based search pattern for fast block motion estimation, *IEEE Trans. Circ. Syst. Video Technol.* 12 (5) (2002) 349–355.
- [48] M. Cagnazzo, F. Castaldo, T. André, M. Antonini, M. Barlaud, Optimal motion estimation for wavelet video coding, *IEEE Trans. Circ. Syst. Video Technol.* 17 (7) (2007) 907–911.
- [49] H. Lakshman, H. Schwarz, T. Blu, T. Wiegand, Generalized interpolation for motion compensated prediction, in: Proceedings of the IEEE International Conference on Image Processing, Brussels, Belgium, 2011.
- [50] P.R. Hill, T.K. Chiew, D.R. Bull, C.N. Canagarajah, Interpolation free subpixel accuracy motion estimation, *IEEE Trans. Circ. Syst. Video Technol.* 16 (12) (2006) 1519–1526.
- [51] M.H. Chan, Y.B. Yu, A.G. Constantinides, Variable size block matching motion compensation with applications to video coding, *IEE Proc. Commun. Speech Vis.* 137 (4) (1990) 205–212.
- [52] S.Y. Yap, J.V. McCanny, A VLSI architecture for variable block size video motion estimation, *IEEE Trans. Circ. Syst. Express Briefs* 51 (7) (2004) 384–389.
- [53] C.-Y. Chen, S.-Y. Chien, Y.-W. Huang, T.-C. Chen, T.-C. Wang, L.-G. Chen, Analysis and architecture design of variable block-size motion estimation for H.264/AVC, *IEEE Trans. Circ. Syst. Regul. Pap.* 53 (3) (2006) 578–593.
- [54] G. de Haan, P.W.A.C. Biezen, H. Huijgen, O.A. Ojo, True-motion estimation with 3-D recursive search block matching, *IEEE Trans. Circ. Syst. Video Technol.* 3 (5) (1993) 368–379.
- [55] I. Moccagatta, F. Moscheni, M. Schütz, F. Dufaux, A motion field segmentation to improve moving edges reconstruction in video coding, in: Proceedings of the IEEE International Conference on Image Processing, Austin, TX, 1994.

- [56] E.M. Hung, R.L. de Queiroz, D. Mukherjee, On macroblock partition for motion compensation, in: Proceedings of the IEEE International Conference on Image Processing, Atlanta, GA, 2006.
- [57] O. Divorra Escoda, P. Yin, C. Dai, X. Li, Geometry-adaptive block partitioning for video coding, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, Hawaii, 2007.
- [58] S. Kamp, M. Evertz, M. Wien, Decoder side motion vector derivation for inter frame video coding, in: Proceedings of the IEEE International Conference on Image Processing, San Diego, CA, 2008.
- [59] G. Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Anal. Mach. Intell.* 7 (4) (1985) 384–401.
- [60] M. Pardas, P. Salembier, B. Gonzalez, Motion and region overlapping motion estimation for segmentation-based video coding, in: IEEE Proceedings of the International Conference on Image Processing (ICIP), Austin, TX, 1994.
- [61] Y.T. Tse, R.L. Baker, Global zoom/pan estimation and compensation for video compression, in: IEEE Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toronto, Canada, 1991.
- [62] A. Smolic, M. Hoeynck, J.-R. Ohm, Low complexity global motion estimation from P-frame motion vectors for MPEG-7 applications, in: IEEE Proceedings of the International Conference on Image Processing (ICIP), Vancouver, Canada, 2000.
- [63] M. Tok, A. Glantz, M.G. Arvanitidou, A. Krutz, T. Sikora, Compressed domain global motion estimation using the Helmholtz tradeoff estimator, in: IEEE Proceedings of the International Conference on Image Processing (ICIP), Hong Kong, 2010.
- [64] P. Anandan, J.R. Bergen, K.J. Hanna, R. Hingorani, Hierarchical model-based motion estimation, in: M.I. Sezan, R.L. Lagendijk (Eds.), *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, 1993, pp. 1–22.
- [65] M. Hoetter, R. Thoma, Image segmentation based on object oriented mapping parameter estimation, *Signal Process.* 15 (3) (1988) 315–334.
- [66] S.F. Wu, J. Kittler, A differential method for simultaneous estimation of rotation, change of scale and translation, *Signal Process. Image Commun.* 2 (1) (1990) 69–80.
- [67] F. Dufaux, J. Konrad, Efficient, robust, and fast global motion estimation for video coding, *IEEE Trans. Image Process.* 9 (3) (2000) 497–501.
- [68] R. Szeliski, J. Coughlan, Hierarchical spline-based image registration , in: IEEE Proceedings of Computer Vision and Pattern Recognition, Seattle, Washington, 1994.
- [69] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1988.
- [70] F. Moscheni, F. Dufaux, M. Kunt, A new two-stage global/local motion estimation based on a background/foreground segmentation, in: IEEE Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Detroit, MI, 1995.
- [71] P. Meer, D. Mintz, A. Rosenfeld, D.Y. Kim, Robust regression methods for computer vision: a review, *Int. J. Comput. Vis.* 6 (1) (1991) 59–70.
- [72] P.J. Rousseeuw, A.M. Leroy, *Robust Regression and Outlier Detection*, Wiley-Interscience, New York, 1987.
- [73] S. Ayer, P. Schroeter, J. Bigün, Segmentation of moving objects by robust motion parameter estimation over multiple frames, in: Proceedings of European Conference on Computer Vision (ECCV), Stockholm, Sweden, 1994.
- [74] J.M. Odobez, P. Bouthemy, Robust multiresolution estimation of parametric motion models, *J. Vis. Commun. Image Represent.* 6 (4) (1995) 348–365.

- [75] P. Burt, E. Adelson, The Laplacian pyramid as a compact image code, *IEEE Trans. Commun.* 9 (4) (1983) 532–540.
- [76] P. Burt, C. Yen, X. Xu, Multi-resolution flow-through motion analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1983.
- [77] W. Enkelmann, Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences, *Comput. Vis. Graph. Image Process.* 43 (2) (1988) 150–177.
- [78] F. Glazer, Multilevel relaxation in low-level computer vision, in: A. Rosenfeld (Ed.), *Multiresolution Image Processing and Analysis*, Springer, Berlin, 1984, pp. 312–330.
- [79] P. Anandan, A unified perspective on computational techniques for the measurement of visual motion, in: *IEEE Proceedings of the International Conference on Computer Vision*, London, UK, 1987.
- [80] M. Bieling, Displacement estimation by hierarchical block matching, in: *SPIE Proceedings of Visual Communication and Image Processing* Cambridge, MA, November 1988.
- [81] M. Black, P. Anandan, The robust estimation of multiple motions: parametric and piecewise-smooth flow fields, *Comput. Vis. Image Understanding* 63 (1) (1996) 75–104.
- [82] F. Dufaux, I. Moccagatta, B. Rouchouze, T. Ebrahimi, M. Kunt, Motion compensated generic coding of video based on multiresolution data structure, *Opt. Eng.* 32 (7) (1993) 1559–1570.
- [83] A. Bruhn, J. Weickert, T. Kohlberger, C. Schnörr, A multigrid platform for real-time motion computation with discontinuity-preserving variational methods, *Int. J. Comput. Vis.* 70 (3) (2006) 257–277.
- [84] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G.J. Sullivan, Rate-constrained coder control and comparison of video coding standards, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 688–703.
- [85] S. Nogaki, M. Ohta, An overlapped block motion compensation for high quality motion picture coding, in: *IEEE Proceedings of the International Symposium on Circuits and Systems*, San Diego, CA, May 1992, pp. 184–187.
- [86] M.T. Orchard, G.J. Sullivan, Overlapped block motion compensation: an estimation-theoretic approach, *IEEE Trans. Image Process.* 3 (5) (1994) 693–699.
- [87] T. Ebrahimi, F. Dufaux, Y. Nakaya, MPEG-4 natural video – II, in: A. Puri, T. Chen (Eds.), *Multimedia Systems, Standards, and Networks*, Marcel Dekker, 2000, pp. 245–269 (Chapter 9).
- [88] MPEG-4 Video Verification Model Editing Committee, The MPEG-4 Video Verification Model 12.1, ISO/IEC JTC1/SC29/WG11 N2552, Rome, December 1998.
- [89] H. Jozawa, K. Kamikura, A. Sagata, H. Kotera, H. Watanabe, Two-stage motion compensation using adaptive global MC and local affine MC, *IEEE Trans. Circ. Syst. Video Technol.* 7 (1) (1997) 75–85.
- [90] T. Wiegand, E. Steinbach, B. Girod, Affine multipicture motion-compensated prediction, *IEEE Trans. Circ. Syst. Video Technol.* 15 (2) (2005) 197–209.
- [91] A. Glantz, A. Krutz, T. Sikora, Adaptive global motion temporal prediction for video coding, in: *Proceedings of Picture Coding Symposium*, Nagoya, Japan, December 2010.
- [92] A. Krutz, A. Glantz, M. Tok, M. Esche, T. Sikora, Adaptive global motion temporal filtering for high efficiency video coding, *IEEE Trans. Circ. Syst. Video Technol.* 22 (12) (2012).
- [93] M. Irani, P. Anandan, S. Hsu, Mosaic based representations of video sequences and their applications, in: *International Conference on Computer Vision*, 1995.
- [94] R. Szeliski, Video mosaics for virtual environments, *IEEE Comput. Graph. Appl.* 16 (1996) 22–30.
- [95] L.A. Teodosio, W. Bender, Salient video stills: content and context preserved, in: *Proceedings of ACM International Conference on Multimedia* Anaheim, CA, 1993.
- [96] F. Dufaux, F. Moscheni, Background mosaicking for low bit rate video coding, in: *IEEE Proceedings of the International Conference on Image Processing*, Lausanne, Switzerland, 1996.
- [97] M. Irani, S. Hsu, P. Anandan, Mosaic-based video compression, in: *SPIE Proceedings of Digital Video Compression: Algorithms and Technologies*, San Jose, CA, 1995.

- [98] M.C. Lee, W. Chen, C.B. Lin, C. Gu, T. Markoc, S.I. Zabinsky, R. Szeliski, A layered video object coding system using sprite and affine motion model, *IEEE Trans. Circ. Syst. Video Technol.* 7 (1) (1997) 130–145.
- [99] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos, California, 1990.
- [100] Y. Lu, W. Gao, F. Wu, Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques, *IEEE Trans. Circ. Syst. Video Technol.* 13 (5) (2003).
- [101] A. Krutz, A. Glantz, M. Frater, T. Sikora, Rate-distortion optimized video coding using automatic sprites, *IEEE J. Sel. Top. Signal Process.* 5 (7) (2011) 1309–1321.
- [102] A. Smolic, T. Sikora, J.-R. Ohm, Long-term global motion estimation and its application for sprite coding, content description, and segmentation, *IEEE Trans. Circ. Syst. Video Technol.* 9 (8) (1999).
- [103] D. Farin, M. Haller, A. Krutz, T. Sikora, Recent developments in panoramic image generation and sprite coding, in: *IEEE Proceedings of the International Workshop on Multimedia Signal Processing*, 2008.
- [104] R. Szeliski, Prediction error as a quality metric for motion and stereo, in: *Proceedings of the IEEE International Conference on Computer Vision*, 1999.
- [105] D. Scharstein, R. Szeliski, High-accuracy stereo depth maps using structured light, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
- [106] Z. Wang, A.C. Bovik, Mean squared error: love it or leave it? A new look at signal fidelity measures, *IEEE Signal Process. Mag.* 26 (1) (2009) 98–117.
- [107] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [108] H.R. Sheikh, A.C. Bovik, Image information and visual quality, *IEEE Trans. Image Process.* 15 (2) (2006) 430–444.
- [109] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, V. Lukin, On between-coefficient contrast masking of DCT basis functions, in: *Proceedings of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-07)*, Scottsdale, AZ, January 2007.
- [110] ITU-R BT.500-12, Methodology for the Subjective of the Quality of Television Pictures, 2009.
- [111] ITU-R P.910, Subjective Video Quality Assessment Methods for Multimedia Applications, 2008.

## 3

# High Efficiency Video Coding (HEVC) for Next Generation Video Applications

Yanxiang Wang\*, Charith Abhayaratne\*, and Marta Mrak†

\*Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S1 3JD, United Kingdom  
†British Broadcasting Corporation, Research and Development Department, London W12 7SB, United Kingdom

## Nomenclature

<b>B (picture)</b>	coded picture predicted using bi-directional motion compensation
<b>B (slice)</b>	a slice that may be decoded using intra prediction or inter prediction using at most two motion vectors and reference indices to predict the sample values of each block. B stands for bi-predictive
<b>BD-rate/ BD-PSNR</b>	an objective evaluation method used in compression performance metric. BD refers to <b>Bjontegaard Delta</b>
<b>BD-MOS</b>	a subjective evaluation method used in compression performance metric
<b>Cb, Cr</b>	<b>chroma signal</b> , specifying that a sample array or single sample is representing one of the two color ( <b>blue</b> and <b>red</b> ) difference signals related to the primary colors
<b>I (Slice)</b>	a slice that is decoded using intra prediction only
<b>P (Slice)</b>	a slice that may be decoded using intra prediction or inter prediction using at most one motion vector and reference index to predict the sample values of each block
<b>PART_2Nx2N, PART_2NxN, PART_Nx2N, PART_NxN, PART_2NxnU, PART_2NxnD, PART_nLx2N, PART_nRx2N</b>	there are eight possible <b>partition modes</b> for prediction units in <b>Figure 3.4</b> ( $N, U, D, L, R$ , and $n$ are all positive integers)
<b>QP</b>	the value of <b>Quantization Parameter</b> . In HEVC, the range of QPs is from <b>0 to 51</b> . In this chapter, the experiments set four QP values: 22, 27, 32, and 37
<b>Y</b>	<b>luma signal</b> , specifying that a sample array or single sample is representing the monochrome signal related to the primary colors

动机

### 5.03.1 Introduction

*High Efficiency Video Coding (HEVC)* is being developed by the *Joint Collaborative Team on Video Coding* (JCT-VC), which is a collaboration between ITU-T *Video Coding Experts Group* (VCEG) and ISO/IEC *Moving Picture Experts Group* (MPEG). The motivation for development of HEVC comes from the need to compress videos more efficiently, with the target to deliver at half the data rate (*a.k.a.* bit-rate) achieved by H.264/AVC standard [1, 2], while retaining the same objective video quality. H.264/AVC, the best performing video compression standard prior to HEVC, has been widely used since it was published by the Joint Video Team of MPEG and VCEG in 2003. Since then, intensive research activities have resulted in a mature technology capable of underpinning a new video compression standard. Following the Call for Proposals [3] issued in January 2010, the development of the new HEVC standard has been in progress. The draft of the standard released in July 2012 [4] demonstrates the achievement of the bit-rate reduction target set by the JCT-VC. The first version of the standard has been released in 2013 [5], about 10 years after H.264/AVC was released.

While H.264/AVC enabled the distribution of *High-Definition (HD)* video broadcasting, it is expected that HEVC will boost and improve such services. The *International Telecommunications Union* (ITU) has announced a recommendation that represents the new television broadcasting environment, termed *Ultra High-Definition Television (UHDTV)* [6]. Due to its superior compression performance, HEVC will pave the way to UHDTV, enabling a significantly improved viewing experience. The resolution of current *HD television (HDTV)* offers about 2.1 Megapixels ( $1920 \times 1080$  pixels/frame). UHDTV systems will support up to 16 times this resolution. Additionally, basic support for screen content (SC) is another new feature that needs to be addressed by next generation video compression standards. SC, which is widely used in mobile and Internet streaming, refers to an image or video that contains computer generated content in addition to elements that are captured by a camera. Computer generated content typically refers to graphics, games, desktop and mobile screens, animations, etc. Since each new content type has different statistical and visual properties, new video compression solutions are needed to achieve high compression gains for such content.

目标

Image and video compression has been a very active field of research and significant development has been achieved over the last 20 years. Many different coding systems and algorithms have been proposed and some are widely used after being adopted into international standards, such as H.262/MPEG-2 video [7], H.263 [8], MPEG-4 Visual [9], and H.264/AVC. Reviewing video compression history, most digital video coding standards were aimed at optimizing the coding efficiency considering the trade-off between bit-rate and video quality. This means minimizing the bit-rate for a certain level of video quality or maximizing the video quality level within a given bit-rate. Higher resolution video content will be even more demanding so the compression efficiency must be further improved in order to store and transmit the video streams successfully within the given disk space and bandwidth constraints. This, in conjunction with the emergence of new generation content and its related requirements has prompted standardization bodies to develop a new high performance video compression standard—HEVC.

约束

The aim of this chapter is to provide an overview of the new video compression standard, HEVC. The basic HEVC architecture and key coding tools are briefly summarized. Both objective and subjective quality evaluations are reported to show the capabilities of HEVC. The performance of relevant key coding tools with respect to different content types is reported. The remainder of this chapter is organized as follows: Section 5.03.2 reviews previous video compression standards and discusses new content

types including UHDTV and SC as well as compression requirements. Section 5.03.3 provides a brief overview of coding tools in the first version of HEVC based on the HEVC text specification draft 8 [4] followed by a performance evaluation and experimental analysis presented in Section 5.03.4. Finally, Section 5.03.5 concludes the chapter and indicates future research directions.

## 5.03.2 Requirements and solutions for video compression

Video applications have a history of innovations in quality starting from monochrome TV and then moving to color TV, HDTV, and now UHDTV. Due to the advances in consumer electronics, the affordability of the equipment, and the convergence of the telecommunications networking, computing, and television technologies, there has been a growth in high-quality user generated content demanding higher resolutions and new hybrid formats such as screen content. Both high-definition video and screen content demand higher qualities and these requirements have driven the features and performance of HEVC. Before looking in detail at HEVC we will first examine the history that has underpinned its development.

### 5.03.2.1 Video compression history

Previous video compression standards have been mainly developed by ITU-T and ISO/IEC. The ITU-T VCEG produced H.261 and H.263; ISO/IEC MPEG produced MPEG-1 and MPEG-4 Visual. These two standardization organizations also jointly cooperated in developing H.262/MPEG-2 Visual, H.264/MPEG-4 AVC, and now HEVC.

**H.261** [10], published in 1988 by ITU-T, was the first truly practical digital video coding standard. It was originally designed to be able to operate at video bit-rates between 40 kbit/s and 2 Mbit/s, supporting CIF (with a resolution of  $352 \times 288$ ) and QCIF (with a resolution of  $176 \times 144$ ) video frame sizes in the 4:2:0 sampling format. The standard supports integer-accuracy motion compensation. H.261 was the first standard to adopt macroblocks. These contain coded data corresponding to a  $16 \times 16$  sample region of the video frame and remained a key feature in all subsequent standards. In the 4:2:0 format, each coded picture consists of a number of macroblocks that contain  $16 \times 16$  luma samples (Y) and  $8 \times 8$  chroma samples (both Cr and Cb). ITU-T then developed the H.263 [8] standard to improve the compression performance at low bit-rates (below 30 kbit/s). H.263 introduced half-pixel motion compensation together with improved motion vector coding which used the reconstructed neighboring blocks as the predictors. The first version of H.263 contained four annexes and version 2 added several optional coding modes such as a deblocking filter and an advanced intra coding mode.

H.262/MPEG-2 Video [7] (widely known as MPEG-2) was the first video coding standard jointly developed by ITU-T VCEG and ISO/IEC MPEG. It was developed as a compression system for digital multimedia content. It is an extension of the prior MPEG-1 video standard with support for interlaced video coding, aimed at *Standard Definition Television* (SDTV) and HDTV coding. Video content was coded at lower resolutions than standard television broadcasting until MPEG-2 was standardized in 1995. It is still widely used for the transmission of digital TV signals over satellite, cable, and terrestrial distribution systems and for the storage of high-quality SD video signals on DVDs. MPEG-2 also uses macroblocks with a maximum block size of  $16 \times 16$ . The standard defines three picture types: I, P, and B. The macroblocks in an I (Intra) picture are coded as a block in a still image without using any prediction with respect to reference frames. The macroblocks in a P (Predicted) picture are coded using either an

操作对象  
resolution

单位  
组成

改进

对象  
支持

单位  
描述

I type macroblock or using forward inter prediction using at most one motion vector and a reference frame to predict the sample values of each block. Macroblocks in a B (Bi-predictive) picture are coded either as an I type, P type or using bi-directional prediction (forward and backward prediction) from two frames on either side of the current frame. Motion compensated prediction is used to form residuals, i.e., the differences between the original and the predicted blocks. MPEG-2 supports motion compensation with half-pixel precision motion vectors. MPEG-2 uses the  $8 \times 8$  Discrete Cosine Transform (DCT) transform to decorrelate the data in I blocks and the prediction residuals in P and B blocks. The quantized transform coefficients are zig-zag scanned and coded using Variable-Length Coding (VLC).

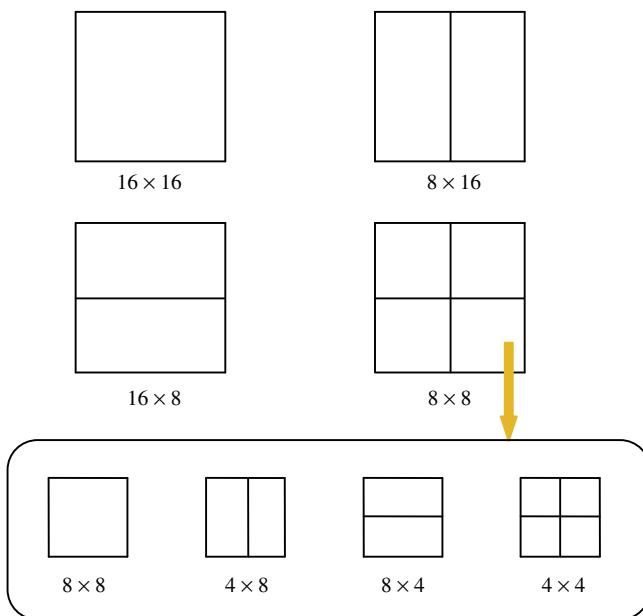
MPEG-4 [9] was developed as an improvement upon MPEG-2. Similar to MPEG-2, the inter prediction can be done with  $16 \times 16$  or  $8 \times 8$  blocks. MPEG-4 added support for motion compensation with quarter-pixel precision motion vector and bilinear interpolation, but it does not specify a deblocking filter in the motion compensation loop as in H.263. The transform in MPEG-4 is similar to previous standards, using the  $8 \times 8$  DCT-based integer transform. The quantized transform coefficients are zig-zag scanned and coded using three-dimensional run-level Variable-Length Coding (VLC).

The aim of H.264/AVC was to create a standard capable of providing good video quality at substantially lower bit-rates than those in the previous MPEG-2, MPEG-4, and H.263 standards. It is the second video compression standard developed jointly by ITU-T and ISO/IEC. The technical characteristics of both compression systems are illustrated in Table 3.1.

H.264/AVC uses a macroblock-based coding structure. The maximum size of coding unit defined in H.264/AVC is  $16 \times 16$  pixels. Each picture is partitioned into  $16 \times 16$  macroblocks, which can be further split into smaller blocks. The luma component of each macroblock may be split up in four ways as shown in Figure 3.1. Directional intra prediction in the spatial domain is used in order to enhance prediction leading to reduced residual energy. Different prediction block sizes are used depending on the content statistics in the region covered by the macroblock:  $16 \times 16$  intra prediction is used in highly homogeneous blocks, while  $4 \times 4$  intra prediction is used in regions containing fine details, i.e., high spatial frequency components. In motion compensated prediction, both past and future frames can be chosen as references from which to predict the current frame. Instead of I, P, and B pictures, H.264/AVC defines five different types of picture slices: I, P, B, Switching P (SP), and Switching I (SI). I, P, and B slices are the same as in previous standards. An SP slice contains P and/or I type macroblocks switching between encoded streams. An SI slice contains a special type of I type macroblock switching between

**Table 3.1** Technical Characteristics of Previous Video Standards

Algorithm characteristic	MPEG-2 and MPEG-4	H.264/AVC
Coded image types	I, B, P	I, B, P, SP, SI
Transform	$8 \times 8$ DCT	$4 \times 4$ and $8 \times 8$ DCT
Motion estimation blocks	$16 \times 16$	$16 \times 16, 8 \times 8, 8 \times 4, 4 \times 4$
Entropy coding	VLC	CAVLC and CABAC
Fractional Motion Estimation	1/2 Pixel (MPEG-2) 1/4 Pixel (MPEG-4)	1/4 Pixel
Deblocking filter	None	Yes

**FIGURE 3.1**

Macroblock partitioning in H.264/AVC.

coded streams. A picture may consist of different types of slices, each containing an integer number of macroblocks. In H.264/AVC,  $4 \times 4$  and  $8 \times 8$  DCT-based integer transforms are used for decorrelating the prediction residuals prior to quantization and entropy coding. Typically, the smaller transform block size is used in rich texture regions, while the larger transform block size is used in smooth regions.

**使用规律**  
**几种技术**

Entropy coding in H.264/AVC is primarily performed by using Context Adaptive Variable-Length Coding (CAVLC). Alternatively, Context Adaptive Binary Arithmetic Coding (CABAC) can be used for enhanced performance. CAVLC is based on Huffman coding principles and uses adaptively generated coding contexts for encoding the symbols efficiently. CABAC converts symbols into binary codes and uses binary arithmetic coding dependent on the coding context. In addition, H.264/AVC also uses an interim processing step, known as in-loop filtering, within the motion compensated inter prediction steps. The in-loop filter is applied on the reconstructed pixels after inverse quantization, the inverse transform, and the synthesis. It is applied in a form of a deblocking filter adaptively at block boundaries to remove the blocking artifacts. This has led to improved rate-distortion performance.

### 5.03.2.2 Next generation formats and content

Since earlier video compression standards were optimized for lower resolutions such as QCIF and CIF, they are unlikely to achieve the best performance for HD resolutions. HD content is now widely used in television broadcasting and cinema, on mobile phones, and for Internet streaming. Higher definition

xx约束下  
解决xx

video content will require more storage space. Therefore, compression efficiency must be improved in order to save storage space and transmit the video streams successfully within the limited space and bandwidth.

#### 5.03.2.2.1 Ultra high-definition TV (UHDTV)

UHDTV defines two video resolutions, up to  $7680 \times 4320$  pixels/frame. It provides a wider viewing angle both horizontally and vertically than conventional HDTV. UHDTV is specified in the *Radio communication Sector of ITU* (ITU-R) Recommendation BT.2020 (or Rec. 2020) [6]. Comparing with HDTV, which is specified in Rec. ITU-R BT.709 [11], UHDTV contains many advanced features. Rec. 2020 defines two resolutions of  $3840 \times 2160$  and  $7680 \times 4320$ , while Rec. 709 only includes the  $1920 \times 1080$  resolution. The resolutions in both systems have an aspect ratio of 16:9 and use square pixels. UHDTV has a viewing angle of  $100^\circ$  due to the large size of the picture that gives a better viewing experience compared to HDTV, which has a viewing angle of  $30^\circ$ . HDTV specifies the following frame rates: 60, 50, 30, 25, and 24 Hz. For the 60, 30, and 24 Hz systems, picture rates having those values divided by 1.001 are also specified. UHDTV extends these frame rates up to 120 Hz. Only progressive scanning is allowed in UHDTV which means all lines of each frame are drawn in sequence. This is in contrast to interlaced video as used in traditional analog television systems where the odd lines and the even lines of each frames are drawn alternately. UHDTV systems can reproduce colors that cannot be shown with the HDTV color space [12]. Table 3.2 shows the different characteristics between UHDTV and HDTV.

Super Hi-Vision (SHV) [13] is the first system that demonstrates UHDTV functionalities. SHV has been developed by Japan Broadcasting Corporation (NHK, Tokyo, Japan) as a future broadcast system that will give viewers a much greater sensation of reality increasing the quality of experience. Another feature of SHV is that it adopts a 22.2 multi-channel three-dimensional sound system that provides an immersive experience. A demonstration of SHV took place during the London 2012 Olympic Games [14].

**Table 3.2 Characteristics of UHDTV and HDTV**

Parameter	UHDTV	HDTV
Number of pixels	$7680 \times 4320$ $3840 \times 2160$	$1920 \times 1080$
Aspect ratio	16:9	16:9
Standard viewing angle (horizontal)	$100^\circ$	$30^\circ$
Frame frequency (Hz)	120, 60, 60/1.001, 50, 30, 30/1.001, 25, 24, 24/1.001	60, 60/1.001, 50, 30, 30/1.001, 25, 24, 24/1.001
Scanning	Progressive	Interlaced Progressive
Sampling lattice	Orthogonal	Orthogonal
Pixel aspect ratio	1:1 (square pixels)	1:1 (square pixels)

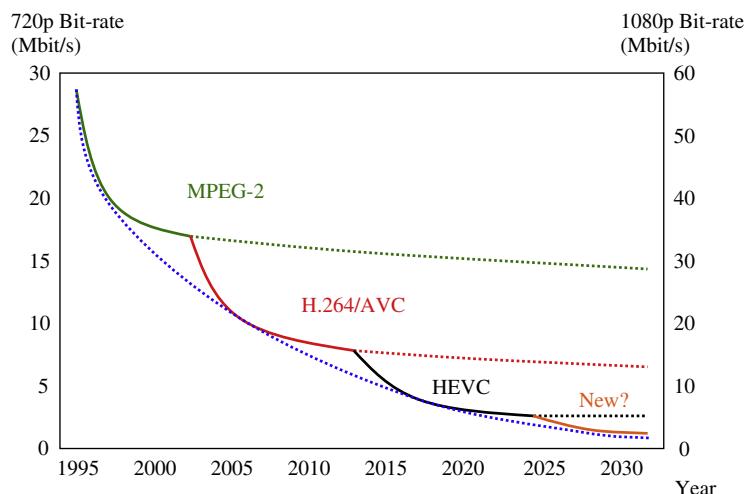
Since UHDTV video has larger frame sizes and rates, more efficient compression methods are needed to improve the compression ratio while maintaining a high quality. [Section 5.03.3](#) will introduce improved coding tools in HEVC that provide better video coding performance.

### 5.03.2.2.2 Screen content

The term **screen content** [15, 16] refers to various types of visual content that exhibits different properties to conventional camera captured content. **Camera captured content**, often known as **natural images**, consists of **homogeneous regions** as well as **texture regions**. It also inherits the camera sensor noise. However, for **non-camera captured content**, such as graphics, the natural texture content may be low, and thus less detail is present. On the other hand for content such as text, there may be frequent **sharp edges** leading to fine details. Depending on the actual content, the spatial activity and temporal activity levels may **vary** in different spatio-temporal regions. Therefore, the blind use of standard coding tools, like **fixed block partitions**, fixed prediction structures, or fixed coding strategies, does not result in an optimum rate-distortion performance nor does it enable delivery of the highest quality required by users. In this case, screen content requires a highly **content-adaptive approach** in all aspects of video coding.

### 5.03.2.3 New market requirements

With ultra high-definition video, the resolution of videos could be more than 30 Megapixels per frame. The need for an improved transmission system is more pronounced and it has to be supported by higher compression efficiency. HEVC, as the successor to H.264/AVC, **aims to halve the bit-rate for a given level of visual quality compared to that achieved by its predecessor**. [Figure 3.2](#) shows the video compression development trend from MPEG-2 to H.264/AVC to HEVC, with bit-rates as suggested



**FIGURE 3.2**

Improvements in coding performance introduced by new standards.

组成  
问题

结果

结论

in [17]. Both H.264/AVC and HEVC have achieved the final goal of halving the bit-rate compared to the previous video compression system. Reviewing the progress of video compression, the development of a compression standard does not always follow a **smooth curve** in reality. The experience of the previous coding system improvements indicates that a significant change of performance can be considered once a decade. Following these trends, HEVC could expect to be used in applications as early as 2015.

### 5.03.3 Basic principles behind HEVC

In this section, the key coding tools in the current HEVC design are presented based on the HEVC text specification draft 8 [4]. Similar to its predecessors, HEVC retains the **block-based hybrid coding** scheme while introducing more **flexible coding tree structures** and **subpartitioning mechanisms**. It also employs **extended directional intra prediction**, **adaptive prediction of motion parameters**, an improved **deblocking filter**, and an **enhanced version of CABAC** [18].

**几个特征**

**单位  
对象**

HEVC relies on dividing a video into multiple **Coding Tree Units (CTUs)**. Each coding process, including inter/intra prediction, quantization, transformation, deblocking, and entropy coding, is performed on a **CTU-basis** or on its **partitions**. Related **coding parameters** (e.g., partitioning structure and coding modes) are defined by an encoder and are used at the reconstruction side. The **intra prediction** process at the encoder aims to select the best directional intra mode among the 35 modes defined in HEVC (Planar, DC, and 33 angular prediction modes). For this process, mode-dependent smoothing of **reference samples** is applied to increase prediction efficiency. The three most probable modes are derived for each block to increase symbol coding efficiency. Additionally, prediction from the luma mode is applied to improve efficiency of chroma mode coding. The inter prediction process uses **Adaptive Motion Vector Prediction (AMVP)** in which motion predictors are selected and coded among several candidates explicitly. Quarter sample luma interpolation is used to derive the prediction. In both intra and inter prediction cases, the **residuals** are generated by subtracting the prediction from the input and then they are spatially **transformed** and **quantized**. The DCT-based transforms are used in both intra and inter modes with the **transform unit** sizes of  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$ , and  $4 \times 4$ . The two-dimensional transform is implemented as two one-dimensional transforms applied horizontally first and then vertically. Additionally, a transform based on the  $4 \times 4$  **Discrete Sine Transform (DST)** is used for intra predicted luma residuals only. Two **in-loop filtering processes** (**deblocking filter** and **sample adaptive offset SAO**) are applied on the **reconstructed** samples to remove blocking artifacts and to reduce the distortion between the original frames and the reconstructed frames. Reconstructed CTUs are assembled to construct a picture. In typical configurations which involve motion compensation, such frames are stored into a decoder picture buffer to be used for prediction of the next picture of input video. HEVC uses CABAC, which relies on binarization, context modeling and binary arithmetic coding, as the single entropy coding method. A comparison of features used in H.262/MPEG-2, H.264/AVC, and HEVC is given in [Table 3.3](#).

The following subsections describe the flexible partitioning of blocks of pixels in HEVC and the transform techniques used on such blocks.

#### 5.03.3.1 Picture partitions

Similar to the techniques used in previous video coding standards, such as H.264/AVC, HEVC uses **adaptive block partitioning**. Additional **flexibility** is introduced by **application of quadtree partitioning**.

**Table 3.3** Comparison of Video Compression Standards

Comparison	H.262/MPEG-2	H.264/AVC	HEVC
Max block size	$16 \times 16$	$16 \times 16$	$64 \times 64$
Intra prediction	None	Up to 9 modes	Up to 35 modes
Motion vector accuracy	Half-pixel	Quarter-pixel	Quarter-pixel
Motion compensation minimum size	$8 \times 8$	$4 \times 4$	$4 \times 4$
In-loop filters	None	Deblocking filter	Deblocking filter Sample adaptive offset
Transforms	$8 \times 8$ DCT	$4 \times 4$ and $8 \times 8$ integer DCT	$4 \times 4 \sim 32 \times 32$ integer DCT $4 \times 4$ integer DST
Entropy coding	VLC	CAVLC CABAC	CABAC

传输方式

Blocks of pixels in a CTU can be recursively split into smaller coding units, which can be further split into prediction units and transform blocks. The decisions for each split are sent in the bit stream.

描述组成范围

#### 5.03.3.1.1 Coding tree unit (CTU)

In HEVC, pictures are divided into a sequence of square CTUs. Typically, a CTU consists of an  $N \times N$  block of single component samples (e.g., luma samples) and two sets of chroma samples (e.g., Cb and Cr). In the 4:2:0 format, each chroma component related to a CTU consists of  $N/2 \times N/2$  samples. In the main profile of the standard, the maximum allowed  $N$  is 64. In contrast to the macroblock structures used in previous standards, such as H.264/AVC, CTUs in HEVC are not just larger but also have different options for their divisions into smaller blocks.

#### 5.03.3.1.2 Coding unit structure

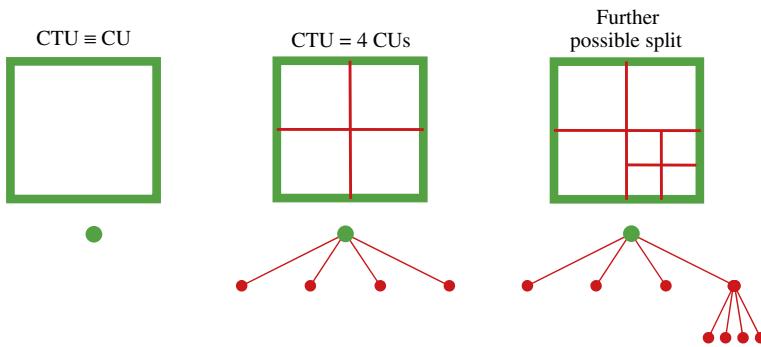
A CTU consists of a number of Coding Units (CUs). A CU consists of square blocks that can be the size of related CTU blocks, or each can be recursively split into four smaller equally sized CUs (up to  $8 \times 8$  luma samples). A CTU splitting into a number of CUs is defined by a quadtree, i.e., such split has a recursive structure. The CU is the basic unit of region splitting used for prediction or transformation. Coding (prediction) inside a CU can be intra or inter. Several examples of CTU splitting into CUs are given in Figure 3.3.

Both skipped CU and non-skipped CU types are allowed in HEVC. The skipped CU is considered in the inter prediction mode without coding of motion vector differences and residual information. The non-skipped CU is assigned to one of two prediction modes, intra prediction and inter prediction.

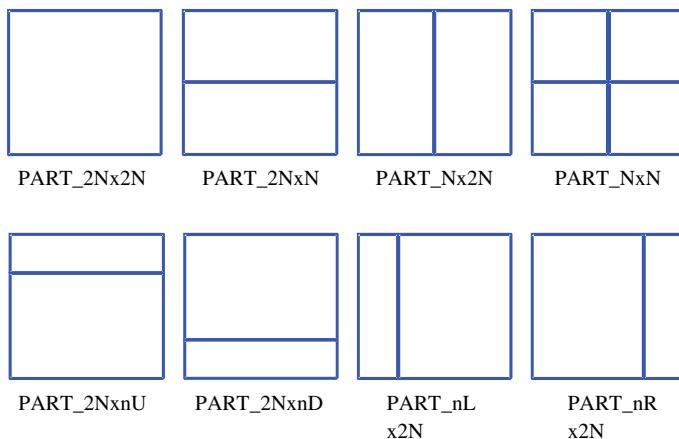
描述，功能

#### 5.03.3.1.3 Prediction unit structure

Each CU may contain one or more Prediction Units (PUs) depending on partitioning modes. The PU is the basic unit used for carrying complete information related to the prediction processes. It is not essential to keep it square in shape, in order to achieve optimal partitioning which matches the

**FIGURE 3.3**

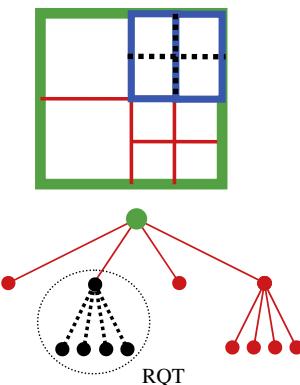
Examples of CTU split into CU and related quadtrees describing splitting.

**FIGURE 3.4**

Partitioning modes for PUs.

后续操作

boundaries of real objects in frames. As illustrated in Figure 3.4, a CU can be split into 1, 2, or 4 PUs. All of the eight possible **partition modes** are used for inter-coded CU. Only **symmetric** PUs, such as PART\_2Nx2N and PART\_NxN partition modes, can be used for **intra-coded** CUs. Partition mode PART\_NxN is allowed only when the corresponding CU size is greater than the minimum CU size. CU with PART\_2Nx2N having only one PU while CU with PART\_NxN has four PUs. The remaining prediction modes have two PUs. In order to reduce the **memory bandwidth** associated with motion compensation, the  $4 \times 4$  block size is not allowed for inter-coded PUs. The main **information** related to each PU are **prediction details**. For example, if a CU is an intra-coded CU then a prediction mode is defined for each PU.

**FIGURE 3.5**

A CTU split with top-right CU split into two PUs using PART\_Nx2N partitioning and one level of TU split (dotted black lines).

约束

#### 5.03.3.1.4 Transform unit structure

The **Transform Unit (TU)** is the basic unit used for the transform and quantization processes. Each CU contains one or more TUs. While the PU split defines prediction in different parts of CU, TU split defines transforms within a CU. Each TU consists of square blocks, ranging from  $32 \times 32$  to  $4 \times 4$  pixels. Similarly to the CTU split into CUs, a CU split into TUs is described by a **quadtree structure**. In this case the split structure is called **Residual QuadTree (RQT)**. Some specific **restrictions** are applied, depending on the prediction mode, PU split, given component (luma or chroma), and block sizes. An example demonstrating the RQT position relative to CTU/PU split is shown in [Figure 3.5](#). Actual transforms and quantization are performed on each TU.

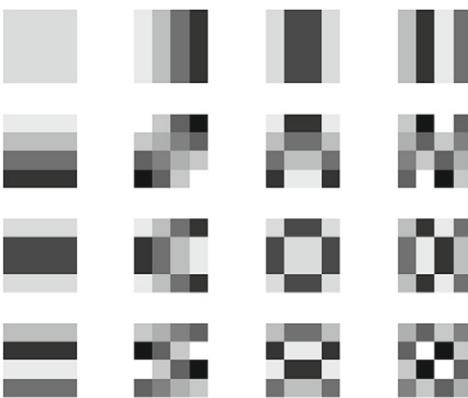
#### 5.03.3.2 Transforms and $4 \times 4$ transform skipping

HEVC specifies **five block transforms** to code prediction residuals:  $4 \times 4$  **DST**-like and four **DCT**-like transforms of sizes  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$ . DST- and DCT-like functions in HEVC use integer arithmetic and have been designed to have basis functions similar to the DCT and DST transforms. From this point in this chapter, related transforms used in HEVC will be simply referred to as DCT and DST transforms. All transforms in HEVC are separable, i.e., they can be applied to the columns and rows of a block separately.

The  $4 \times 4$  DCT transform in HEVC is defined by the following matrix:

$$\begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}.$$

The transform basis patterns for its 2D equivalent are shown in [Figure 3.6](#).

**FIGURE 3.6**

Basis patterns for  $4 \times 4$  DCT.

Similarly, the 8-point DCT transform is defined by the following matrix:

$$\begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix}.$$

The transform basis patterns for its 2D equivalent are shown in Figure 3.7. For the definition of larger transforms in HEVC, see [4].

The DST has some **compression advantages** over the DCT, especially for intra  $4 \times 4$  residuals. When using **intra prediction**, the samples from the upper and left boundaries are used to predict the current pixel. Therefore, the pixel which is close to the top-left corner can be predicted more **accurately** than the pixel away from the reference samples. In other words, **residuals** of pixels which are away from the top-left neighbors will usually be larger than pixels near neighbors. It can be shown that for such residuals the DST is an optimal choice [19].

In HEVC,  $4 \times 4$  DST is only used in intra modes. The transform matrix of 4-point DST is:

$$\begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}.$$

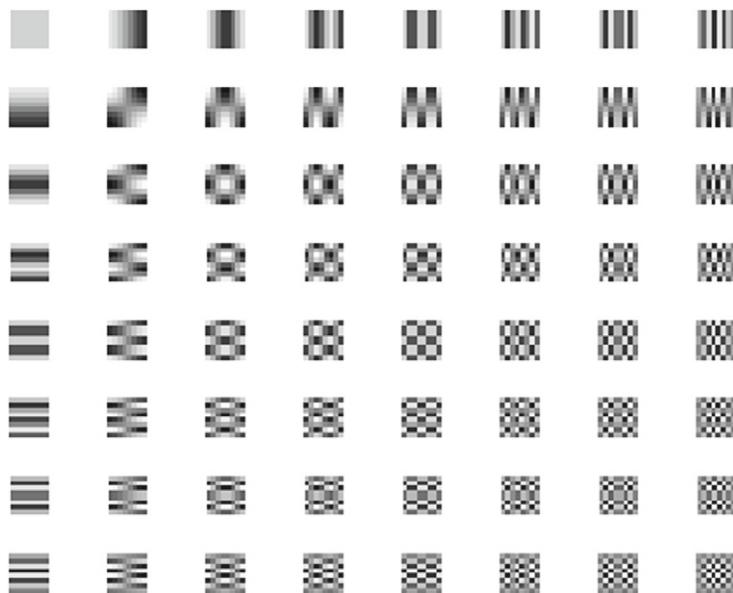
The basis patterns for the  $4 \times 4$  DST are shown in Figure 3.8.

The transform process **concentrates** the energy of the signal into fewer coefficients reducing the correlation in the residual signals. However, for certain types of residuals, the compression efficiency

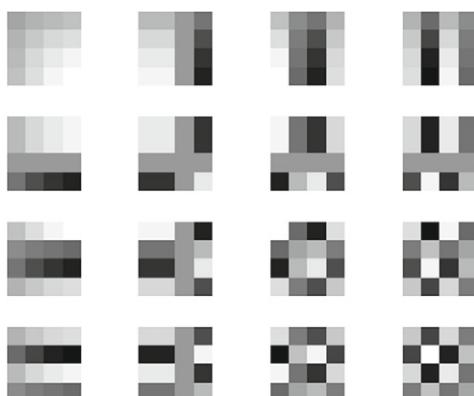
**优势 , 用途**

**准确-残差**

**原理**

**FIGURE 3.7**

Basis patterns for  $8 \times 8$  DCT.

**FIGURE 3.8**

Basis patterns for  $4 \times 4$  DST.

不适用场景

could be improved by completely skipping the transform. For instance, since parts of screen content can be noiseless and usually consist of more regular and sharper signals, intra prediction might work well and lead to much smaller or even zero residuals, compared to coding of camera captured content. In cases where sharp lines are preserved in the residuals, a transform might be inefficient and might

even decrease the compression efficiency. Transform skipping also benefits inter coding, especially for motion compensated residuals which more often consist of sharp edges and uncorrelated pixels, compared to those in intra-predicted blocks. To address such coding features, HEVC enables full skip of the 2D transform for  $4 \times 4$  blocks. In the presence of **transform skipping**, all other processing steps remain unchanged. However, additional normalization of signals is required in order to preserve effective application of unchanged quantization and entropy coding.

In [Section 5.03.4](#), the performance improvements introduced by application of transform skipping in HEVC are demonstrated.

### 5.03.4 Performance evaluation

This section presents a performance evaluation of HEVC and a comparison with that for H.264/AVC. Additionally, the features of HEVC that introduce significant gains for coding of high resolution and screen content are evaluated. While the detailed results presented here are based on an objective evaluation study, it should be noted that limited subjective test results reported in the literature already indicate superior subjective quality of HEVC, compared to H.264/AVC [20].

In objective evaluations, the **bit-rate reduction** is typically calculated using *Bjontegaard Delta rate (BD-rate)* metric [21]. The **distortion** can be measured using *Bjontegaard Delta PSNR (BD-PSNR)* metric and *Bjontegaard Delta Mean Opinion Score (BD-MOS)* metric for objective and subjective evaluations, respectively.

The JCT-VC common test conditions define a set of configurations [22] used in experiments during the development of HEVC. Common test conditions are desirable to conduct experiments in a well-defined environment and ease the comparison of the outcome of experiments. The **configurations** used in our experiments include *All Intra* (AI), *Random Access* (RA), and *Low Delay with B slices* (LDB). In AI, all frames are coded by using only one slice. The RA configuration is typically applied in a **broadcasting** environment, which uses pyramidal picture reordering with a random access picture approximately every 1 second. The LDB configuration is often used in a **video conferencing** where **picture reordering is not applied** and only the **first frame is encoded as an I slice**. Four *Quantization Parameters (QP)* 22, 27, 32, and 37 are used for intra-coded frames while the QPs for the inter ones are automatically derived by the encoder according to the frame position in the *Group of Picture (GOP)* and the GOP length.

#### 5.03.4.1 Subjective quality evaluation

A subjective quality study reported in [20] involved 36 subjects that visually examined three complex video sequences compressed using both H.264/AVC and HEVC. The three video sequences, namely, People on street ( $3840 \times 2160$ , 30 fps), Traffic ( $3840 \times 2048$ , 30 fps), and Sintel2 ( $3840 \times 1744$ , 24 fps), are all near 4K resolution with frame rates of either 24 fps or 30 fps.

The video sequences are encoded at various bit-rates in a Random Access configuration. Considering the different content features and different spatio-temporal characteristics, the bit-rates are set separately for each sequence. The research uses ***The Double Stimulus Impairment Scale (DSIS)*** method which presents video sequences continuously to subjects and asks them to rate visual quality on a rating scale with a maximum score of 100. The distance between the monitor and the subjects was set approximately **3.5 times the height of the screen**.

配置  
不同的环境  
/使用场景

原理

参数

Some sequences used in the visual test are relatively hard to encode because of related higher spatial information and temporal information indexes. For example in PeopleOnStreet the content was smoothed in the HEVC reconstructed sequences while blocking was perceived in H.264/AVC. For this content, HEVC achieved a high bit-rate (up to 74%) reduction. The sequence Traffic is easier to encode but also benefits from high bit-rate reductions introduced by HEVC. In subjective tests BD-MOS values of HEVC over H.264/AVC range from 51% to 74%. The corresponding BD-PSNR values range from 28% to 68%.

#### 5.03.4.2 Objective quality evaluation

We evaluated the performance of the HEVC test model version 8.0 (HM-8.0) and compared it with that of the high-profile H.264/AVC standard (JM 18.4 software). All JCT-VC test video sequences specified in [22] were used in tests, including Class F, which consists of the screen content video. In the test that compares H.264/AVC and HEVC performance, six configurations were used for HEVC: AI-Main, RA-Main, LDB-Main, AI-HE, RA-HE, and LDB-HE, where **HE** stands for **High Efficiency**. The High Efficiency setting defined in [22] differs from the Main setting in the internal bit-depth used. While the Main setting used 8-bit **internal processing precision**, the HE setting uses 10-bit **internal bit-depth**. For evaluations of different HEVC settings, only the Main configurations were used.

The JCT-VC common test condition provides six classes of test sequences. In general, Class A to E test sequences are **camera captured content** and Class F contains **screen content sequences**. All sequences are in 4:2:0 chrominance sampling **format**. There are four Class A sequences with the resolution of  $2560 \times 1600$  pixels and 30 and 60 fps. Class A sequences have the highest resolution among all JCT-VC test sequences. Classes B (5 sequences,  $1920 \times 1080$ , 24–60 fps), C (4 sequences,  $832 \times 480$ , 30–60 fps), and D (4 sequences,  $416 \times 240$ , 50 and 60 fps) are tested in all configurations. Class E is video conferencing material which is not tested in RA configurations. All three sequences ( $1280 \times 720$ , 60 fps) are tested in AI and LDB configurations. Class F ( $832 \times 480$ – $1280 \times 720$ , 20–50 fps) contains four sequences that combined both camera captured and text/graphical content, such as graphical overlays, a computer game, and computer desktop content.

##### 5.03.4.2.1 Comparison of HEVC and H.264/AVC

A summary of BD-rate results of the luma component for all JCT-VC test sequences comparing HEVC performance (Main setting) to the performance of H.264/AVC is listed in [Table 3.4](#). Negative values indicate bit-rate savings. [Table 3.5](#) summarizes the BD-rate average results of the evaluation also including results for High Efficiency settings. For screen content (Class F), HEVC achieves gains in the range of 28.3–35% BD-rate, the largest gains are observed for the low-delay configuration. Due to the new coding tools used in HEVC (adaptive large block partition, motion compensation, and SAO loop filter), frames can be more accurately predicted from inter coded frames, and therefore the gain is mostly related to inter coding configurations (RA and LDB). In the High Efficiency settings, gains are larger confirming the benefits of using higher internal bit-depth processing settings.

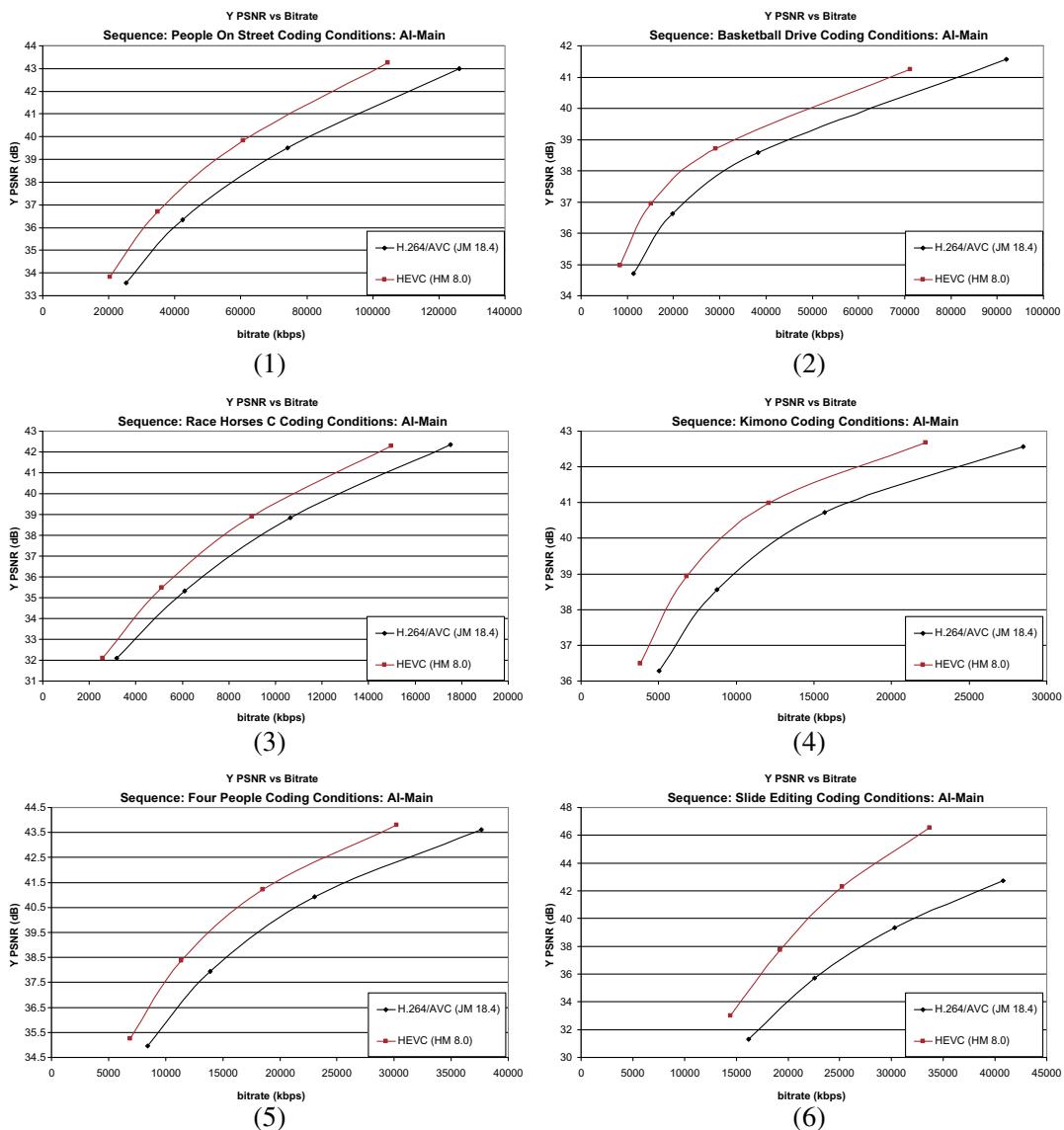
To demonstrate the PSNR and bit-rate ranges involved in BD-rate computations, sequences are selected from each Class (from A to F) providing a selection of content with different resolution and frame rates. [Figure 3.9](#) shows the rate-distortion (RD) curves for the selected test sequences from each Class in AI-Main configuration.

**Table 3.4** BD-rate Results of Luma Component for HEVC Compared to H.264/AVC

Class	Sequence name	Resolution	Bit-depth	Frame rate (fps)	All intra-main (%)	Random access-main (%)	Low-delay-B main (%)
A	Traffic	2560 × 1600	8	30	-21.6	-36.8	N/A
A	People on street	2560 × 1600	8	30	-22.4	-23.4	N/A
A	Nebuta	2560 × 1600	10	60	-25.8	-30.6	N/A
A	Steam Locomotive	2560 × 1600	10	60	-24.8	-56.0	N/A
B	Kimono	1920 × 1080	8	24	-28.6	-44.5	-44.1
B	Park scene	1920 × 1080	8	24	-16.4	-31.5	-34.3
B	Cactus	1920 × 1080	8	50	-22.2	-35.9	-38.2
B	BQ terrace	1920 × 1080	8	60	-19.4	-44.1	-49.5
B	Basketball drive	1920 × 1080	8	50	-26.9	-42.9	-44.9
C	Race horses	832 × 480	8	30	-17.3	-25.3	-27.5
C	BQ mall	832 × 480	8	60	-19.5	-32.4	-33.2
C	Party scene	832 × 480	8	50	-11.6	-28.3	-31.8
C	Basketball drill	832 × 480	8	50	-30.3	-35.0	-38.1
D	Race horses	416 × 240	8	30	-18.4	-23.2	-24.6
D	BQ square	416 × 240	8	60	-12.9	-38.3	-40.4
D	Blowing bubbles	416 × 240	8	50	-12.7	-23.9	-26.6
D	Basketball pass	416 × 240	8	50	-21.7	-25.9	-27.3
E	Four people	1280 × 720	8	60	-23.3	N/A	-34.5
E	Johnny	1280 × 720	8	60	-34.7	N/A	-52.8
E	Kristen and sara	1280 × 720	8	60	-28.6	N/A	-44.6
F	Basketball drill text	832 × 480	8	50	-27.1	-32.9	-37.0
F	China speed	1024 × 768	8	30	-27.6	-32.0	-34.4
F	Slide editing	1280 × 720	8	30	-28.3	-27.7	-31.8
F	Slide show	1280 × 720	8	20	-30.4	-30.5	-31.6

**Table 3.5** BD-rate Average Results for Luma Component for HEVC Compared to H.264/AVC

Class	AI-main (%)	AI-HE (%)	RA-main (%)	RA-HE (%)	LDB-main (%)	LDB-HE (%)
A	-23.7	-25.2	-36.7	-38.1	N/A	N/A
B	-22.7	-23.8	-39.8	-41.0	-42.2	-43.5
C	-19.7	-20.2	-30.3	-31.0	-32.6	-33.3
D	-16.4	-16.8	-27.8	-28.3	-29.7	-30.2
E	-28.9	-30.2	N/A	N/A	-44.0	-46.4
F	-28.4	-28.3	-30.8	-30.9	-33.7	-32.9

**FIGURE 3.9**

Comparison of HEVC and H.264/AVC coding performance (all intra). (1) People on street (Class A), (2) Basketball drive (Class B), (3) Race horses (Class C), (4) Kimono (Class D), (5) Four people (Class E), (6) Slide editing (Class F).

#### 5.03.4.2.2 Coding with reduced block sizes

The results from the previous subsection indicate that HEVC achieves larger gains on higher resolution videos (Class A, B, and E) and smaller gains on lower resolutions (Class C and D). This behavior comes partially from the application of larger block sizes whose performance is evaluated in this subsection.

Under HEVC common test conditions, each frame is split into a number of CTUs with luma block size of  $64 \times 64$  pixels. Each CTU can be recursively split into smaller CUs depending on the applied coding conditions and rate-distortion decisions. In order to evaluate the influence of different block sizes, two tests were performed. While keeping the common test conditions settings with CTUs with  $64 \times 64$  blocks, in the first test the maximal CU size was limited to  $16 \times 16$  pixels (equivalent to macroblock size in H.264/AVC), while in the second test the maximal CU size was limited to  $8 \times 8$  pixels. The results are compared with HEVC results reported in the previous section (where maximal CU size equals CTU size).

The results obtained are summarized in [Table 3.6–3.9](#). Positive BD-rate values indicate a bit-rate increase introduced by the limitation of block sizes available for coding. The results show that the

**Table 3.6** BD-rate Results for Luma Component for HEVC with Maximal Sizes of CU of  $16 \times 16$  Pixels, Compared to Typical HEVC Settings with Maximal CU Size of  $64 \times 64$  Pixels

Class	Sequence name	All intra-main (%)	Random access-main (%)	Low-delay-B main (%)
A	Traffic	1.1	13.5	N/A
A	People on street	0.7	3.0	N/A
A	Nebuta	5.3	11.7	N/A
A	Steam locomotive	7.0	56.4	N/A
B	Kimono	5.8	21.8	16.6
B	Park scene	1.1	10.7	9.2
B	Cactus	1.3	10.6	9.3
B	BQ terrace	0.9	14.9	18.3
B	Basketball drive	2.7	18.8	16.4
C	Race horses	0.9	5.7	4.6
C	BQ mall	0.8	8.7	8.6
C	Party scene	0.1	4.3	4.0
C	Basketball drill	0.7	8.4	8.7
D	Race horses	0.5	2.8	2.2
D	BQ square	0.3	4.5	4.8
D	Blowing bubbles	0.0	3.2	3.4
D	Basketball pass	1.0	4.1	3.8
E	Four people	1.2	N/A	16.0
E	Johnny	6.6	N/A	50.0
E	Kristen and Sara	3.5	N/A	32.8
F	Basketball drill text	0.5	6.6	6.8
F	China speed	0.7	6.1	6.3
F	Slide editing	0.5	4.1	12.9
F	Slide show	3.2	9.7	11.9

**Table 3.7** Average BD-rate for Luma Component for HEVC with Maximal Sizes of CU of  $16 \times 16$  Pixels, Compared to Typical HEVC Settings with Maximal CU Size of  $64 \times 64$  Pixels

Class	All-main (%)	RA-main (%)	LDB-main (%)
A	3.5	21.1	N/A
B	2.4	15.4	14.0
C	0.6	6.8	6.5
D	0.5	3.7	3.5
E	3.8	N/A	32.9
F	1.2	6.7	9.5

**Table 3.8** BD-rate Results for Luma Component for HEVC with Maximal Sizes of CU of  $8 \times 8$  Pixels, Compared to Typical HEVC Settings with Maximal CU Size of  $64 \times 64$  Pixels

Class	Sequence name	All intra-main (%)	Random access-main (%)	Low-delay-B main (%)
A	Traffic	7.1	34.7	N/A
A	People on street	5.4	17.8	N/A
A	Nebuta	17.0	34.3	N/A
A	Steam locomotive	28.2	165.8	N/A
B	Kimono	23.2	71.1	63.3
B	Park scene	6.2	29.0	30.4
B	Cactus	7.2	37.7	37.2
B	BQ terrace	4.7	38.3	48.1
B	Basketball drive	14.3	62.7	55.7
C	Race horses	4.6	27.1	23.3
C	BQ mall	4.8	28.0	29.4
C	Party scene	0.8	14.2	17.4
C	Basketball drill	4.2	27.7	28.3
D	Race horses	3.2	17.7	15.1
D	BQ square	1.8	14.6	19.9
D	Blowing bubbles	0.7	12.3	16.1
D	Basketball pass	6.2	19.0	17.8
E	Four people	8.7	N/A	47.8
E	Johnny	24.0	N/A	126.2
E	Kristen and Sara	16.4	N/A	87.8
F	Basketball drill text	3.3	23.0	24.4
F	China speed	4.4	23.2	25.5
F	Slide editing	2.9	10.2	32.4
F	Slide show	16.5	32.8	37.3

**Table 3.9** Average BD-rate for Luma Component for HEVC with Maximal Sizes of CU of  $8 \times 8$  Pixels, Compared to Typical HEVC Settings with Maximal CU Size of  $64 \times 64$  Pixels

Class	AI-main (%)	RA-main (%)	LP-main (%)
A	14.4	63.1	N/A
B	11.1	47.8	46.9
C	3.6	24.3	24.6
D	3.0	15.9	17.2
E	16.4	N/A	87.3
F	6.8	22.3	29.9

compression of JCT-VC sequences with larger resolution significantly benefits from the application of larger blocks for inter coding configurations (RA and LDB).

The rate-distortion curves for a selection of sequences of different resolutions coded in the LDB-Main configuration are shown in [Figure 3.10](#). From the RD curves it is evident that adaptive block partition in HEVC with larger maximal block sizes contributes to efficient compression at all tested bit-rates.

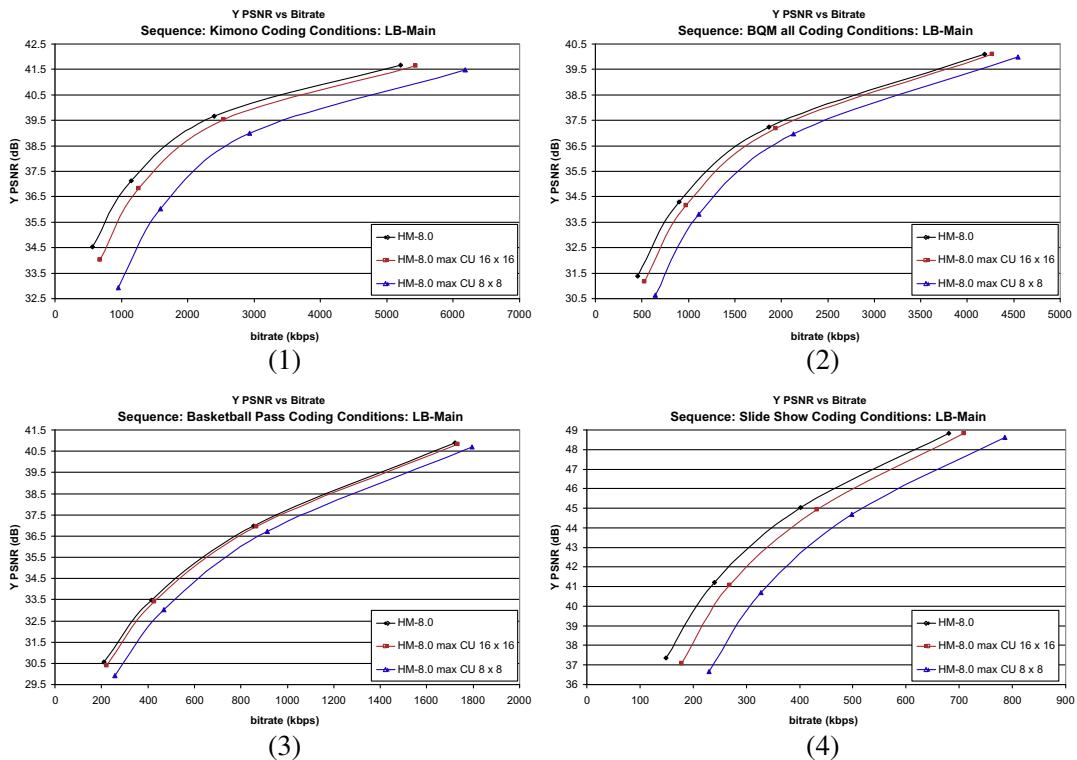
#### 5.03.4.2.3 Transform skipping for screen content

As discussed in [Section 5.03.3.2.](#), transform skipping (TS) can be efficiently used on certain content types, namely, on screen content that contains computer generated elements. In this test the efficiency of transform skipping is evaluated by disabling the normative transform skip option at the encoder, i.e., by forcing application of transforms on each coded block.

While HM-8.0 and common test conditions are used, as in the other evaluations reported in this chapter, the focus of this experiment is on screen content (Class F). However, it should be noted that the effect of disabling transform skip does not lead to significant change in coding performance for Class A–E sequences.

The results for four screen content sequences are summarized in [Table 3.10](#), where HEVC without transform skip is compared to HEVC with transform skip (common test conditions). Positive values of BD-rate indicate losses introduced when transform skip is not an option for compression of given content. The resulting BD-Rate ranged from 0.4 to 16.2% with an average of 8.2%, 7.3%, and 6.4% in All Intra, Random Access, and Low Delay configurations, respectively. It should be noted that the Basketball Drill Text sequence mainly consists of camera-captured content and therefore its coding does not greatly benefit from transform skipping.

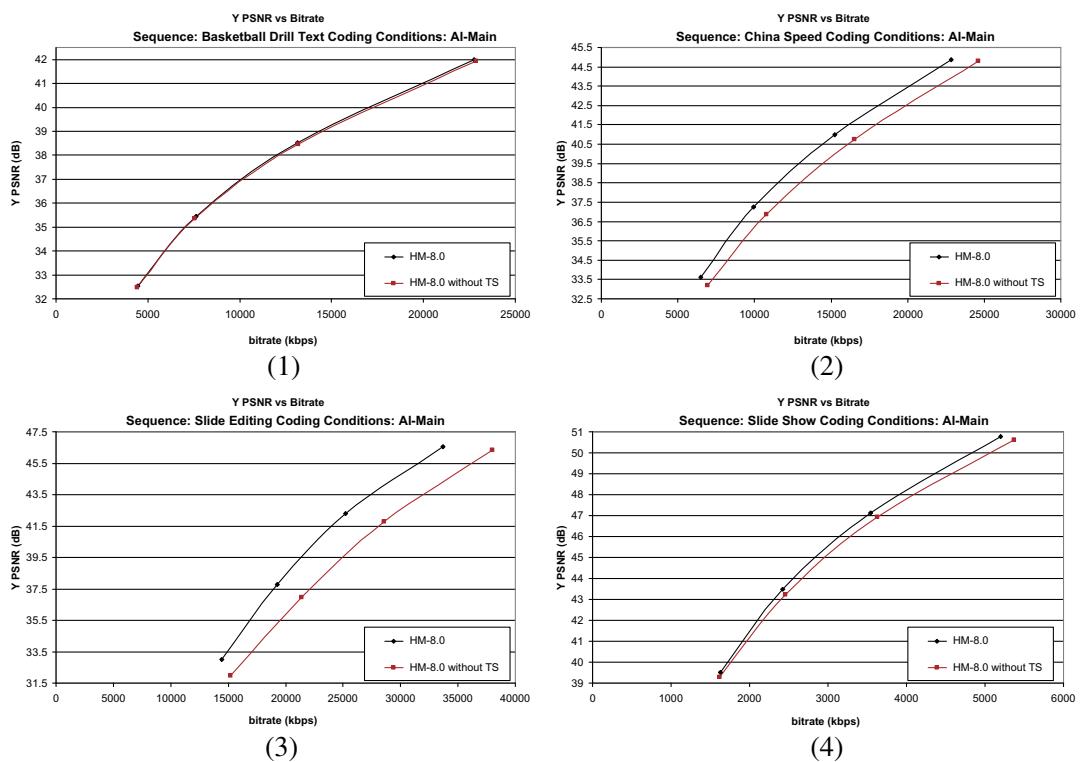
[Figure 3.11](#) shows the RD curves for the four test sequences in AI-Main configuration. The results indicate that the compression efficiency can be improved for screen content when using transform skipping in HEVC.

**FIGURE 3.10**

Comparison of HEVC with different maximal CU sizes on LDB-Main configuration. (1) Kimono (Class B), (2) BQ mall (Class C), (3) Basketball pass (Class D), (4) Slide show (Class F).

**Table 3.10** BD-rate Results for Luma Component for HEVC without Transform Skip Compared to HEVC with Transform Skip (Common Test Conditions)

Class	Sequence name	All intra-main (%)	Random access-main (%)	Low-delay-B main (%)
F	Basketball drill text	0.7	0.6	0.4
F	China speed	11.6	11.1	10.2
F	Slide editing	16.2	13.3	11.5
F	Slide show	4.1	4.3	3.4

**FIGURE 3.11**

Comparison of HEVC without TS and HEVC standard (all intra). (1) Basketball drill text, (2) China speed, (3) Slide editing, (4) Slide show.

### 5.03.5 Conclusions

This chapter introduced the new video compression standard HEVC, which is being developed jointly by ITU-T and ISO/IEC. It also discussed the key features of HEVC and compared it to earlier coding standards. The history of video compression and new market requirements indicate that the development of the HEVC is essential for the next generation video compression technology. High-definition video, especially ultra high-definition resolution, will benefit from the new coding tools available in HEVC.

HEVC applies a more flexible block structure with CTU sizes as large as  $64 \times 64$  pixels and uses quadtree partitioning to recursively divide these into smaller blocks (up to  $8 \times 8$ ). More intra prediction modes (35 compared to 9 in H.264/AVC) and improved motion estimation are also used in HEVC. In-loop processing not only includes an improved deblocking filter but also adds the sample adaptive offset to increase the accuracy of reconstructed images. H.264/AVC uses both the CABAC and the CAVLC as entropy coding methods while HEVC retains enhanced CABAC as the only one.

The experimental results show that HEVC (HM-8.0) improves RD performance for both camera captured video and screen content videos compared to those from H.264/AVC (JM 18.4). The results

show that on average 56% bit-rate saving can be achieved (*e.g.*, Steam Locomotive in RA-Main). Also, the performance of adaptive block partitioning in HEVC has been evaluated to compare coding performance for both small blocks and large blocks. It can be seen that the adaptive block partitioning used in HEVC has better performance than partitioning the frames as smaller sized blocks. When usage of larger blocks is not allowed, the bit-rate consumption increases significantly for content of larger resolution, especially when inter prediction is used. Transform skipping generates significant improvements on screen content while other classes of sequences are not influenced by full transform skipping used in HEVC. By applying transform skipping, bit-rate savings of 0.4–16.2% can be seen for screen content sequences. The performance evaluation results have confirmed the 50% bit-rate saving in HEVC for certain equivalent video quality compared to the prior standards. This is more evident for high-definition video sequences. Research on HEVC extensions is still in progress while the Final Draft International Standard has been released in 2013 [5].

## Relevant Websites

JCTVC-HM software: <http://hevc.kw.bbc.co.uk/git/w/jctvc-hm.git>

Joint Collaborative Team on Video Coding (JCT-VC) <http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx>

The MPEG home page <http://mpeg.chiariglione.org/>

JCT-VC Documents Management System [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/all\\_meeting.php](http://phenix.it-sudparis.eu/jct/doc_end_user/all_meeting.php)

Fraunhofer Heinrich Hertz Institute HEVC website <http://hevc.info/>

ITU-T Recommendation H.265 – High Efficiency Video Coding <http://www.itu.int/rec/T-REC-H.265>

ITU-T Recommendation H.264- Advanced video coding <http://www.itu.int/rec/T-REC-H.264>

IEEE Transactions on Circuits and Systems for Video Technology, Combined Issue on High Efficiency Video Coding (HEVC) Standard and Research <http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=6403920&punumber=76>

## Glossary

### **4:2:0 (sampling)**

a sampling method. Chrominance component has half the horizontal and vertical resolution of luminance component

### **CABAC**

an entropy coding method to reduce redundancy

### **(Context-Based Adaptive Binary**

### **Arithmetic Coding)**

### **CIF (Common Intermediate Format)**

a color image format used to standardize the video sequence with a resolution of  $352 \times 288$

### **CTU (Coding Tree Unit)**

a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples of a picture that has three sample arrays, or a coding tree block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples

<b>CU (Coding Unit)</b>	a coding block of luma samples, two corresponding coding blocks of chroma samples of a picture that has three sample arrays, or a coding block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples
<b>PU (Prediction Unit)</b>	a prediction block of luma samples, two corresponding prediction blocks of chroma samples of a picture that has three sample arrays, or a prediction block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to predict the prediction block samples
<b>QCIF (Quarter Common Intermediate Format)</b>	a color image format used to standardize the video sequence with a resolution of $176 \times 144$
<b>SC (screen content)</b>	non-camera captured images/videos consist of various types of visual content mixed together
<b>TU (Transform unit)</b>	a transform block of luma samples of size $8 \times 8$ , $16 \times 16$ , or $32 \times 32$ or four transform blocks of luma samples of size $4 \times 4$ , two corresponding transform blocks of chroma samples of a picture that has three sample arrays, or a transform block of luma samples of size $8 \times 8$ , $16 \times 16$ , or $32 \times 32$ or four transform blocks of luma samples of size $4 \times 4$ of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to transform the transform block samples

## References

- [1] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 560–576.
- [2] ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC).
- [3] ITU-T, VCEG and ISO/IEC MPEG, Joint Call for Proposals on Video Compression Technology, document VCEG-AM91 and WG11 N11113, Kyoto, Japan, January, 2010.
- [4] B. Bross, W.-J. Han, J.-R. Ohm, G.J. Sullivan, T. Wiegand, High Efficiency Video Coding (HEVC) Text Specifications Draft 8, document JCTVC-J1003 of JCT-VC, Stockholm, Sweden, 2012.
- [5] B. Bross, W.-J. Han, J.-R. Ohm, G.J. Sullivan, T. Wiegand, High Efficiency Video Coding (HEVC) Text Specifications Draft 10 (for EDIS & Last Call), document JCTVC-L1003 of JCT-VC, Geneva CH, January 2013.
- [6] ITU-R, Parameter Values for Ultra-High Definition Television Systems for Production and International Programme Exchange, ITU-R Rec. BT.2020, August 2012.
- [7] ITU-T and ISO/IEC JTC 1, Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2), version 1, 1994.
- [8] ITU-T, Video Coding for Low Bitrate Communication, ITU-T Rec. H.263, version 1, 1995, version 2, 1998, version 3, 2000.
- [9] ISO/IEC JTC 1, Coding of Audio-Visual Objects—Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual), version 1, 1999, version 2, 2000, version 3, 2004.

- [10] ITU-T, Video CODEC for Audiovisual Services at  $p \times 64$  kbit/s, ITU-T Rec. H.261, 1993.
- [11] ITU-R, Parameter Values for High Definition Television Systems for Production and International Programme Exchange, ITU-R Rec. BT.709-5, April 2002.
- [12] Y. Shishikui, K. Iguchi, S. Sakaida, K. Kazui, A. Nakagawa, High-performance video codec for Super Hi-Vision, Proc. IEEE 101 (1) (2013) 130–139.
- [13] S. Sakaida, N. Nakajima, A. Ichigaya, M. Kurozumi, K. Iguchi, Y. Nishida, E. Nakasu, S. Gohshi B, The super Hi-Vision codec, in: Proceedings of the IEEE International Conference on Image Processing, 2007, pp. I.21–I.24.
- [14] J. Zubrzycki, The Olympics in Super Hi-Vision, 2012. <<http://www.bbc.co.uk/blogs/researchanddevelopment/2012/08/the-olympics-in-super-hi-visio.shtml>>.
- [15] T. Lin, P. Hao, Compound image compression for real-time computer screen image transmission, IEEE Trans. Image Proc. 14 (8) (2005) 993–1005.
- [16] H. Meuel, J. Schmidt, M. Munderloh, J. Ostermann, Analysis of coding tools and improvement of text readability for screen content, in: Proceedings of the Picture Coding Symposium (PCS) 2012, May, 2012.
- [17] K. McCann, Progress towards high efficiency video coding, DVB SCENE, March, 2012 p. 5.
- [18] F. Bossen, B. Bross, K. Suehring, D. Flynn, HEVC complexity and implementation analysis, IEEE Trans. Circ. Syst. Video Technol. 22 (12) (2012) 1685–1696.
- [19] C. Yeo, Y.H. Tan, Z. Li, Low-complexity mode-dependent KLT for block-based intra coding, in: Proceedings of the IEEE International Conference on Image Processing 2011, September 2011.
- [20] P. Hanhart, M. Rerabek, F. De Simone, T. Ebrahimi, Subjective quality evaluation of the upcoming HEVC video compression standard, in: Proceedings of the SPIE Optics + Photonics 2012 Applications of Digital Image Processing XXXV, August 2012.
- [21] G. Bjontegaard, Improvements of the BD-PSNR Model, Technical Report VCEG-AI11, ITU-T SG16/Q6, Berlin, Germany, July 2008.
- [22] F. Bossen, Common Test Conditions and Software Reference Configurations, document JCTVC-H1100 of JCT-VC, San Jose CA, USA, February 2012.

# Stereoscopic and Multi-View Video Coding

# 4

**Mark R. Pickering**

*School of Engineering and Information Technology, The University of New South Wales, Canberra, Australia*

---

## Nomenclature

<b>MVC</b>	multi-view coding
AVC	advanced video coding
<b>LCD</b>	liquid crystal display
<b>FTV</b>	free-viewpoint TV
DCT	discrete cosine transform
MV	motion vector
<b>MUX</b>	<b>multiplexor</b>
QP	quantization parameter
DIBR	depth image based rendering
<b>2D + Z</b>	single-view video plus depth
<b>MVD</b>	multi-view video plus depth
<b>LDV</b>	layered depth video
<b>DES</b>	depth enhanced stereo
SSD	sum-of-squared difference
<b>NCC</b>	normalized cross correlation
<b>CGI</b>	computer generated imagery
<b>GBT</b>	graph-based transform
JNDD	just noticeable difference in depth

---

### 5.04.1 Introduction

描述  
实现

目的  
实现  
工具

**Stereoscopic** vision allows humans to perceive objects in three dimensions (3D) by fusing the two slightly different views that are incident on each eye. When viewing images or video displayed on a planar surface, such as a cinema screen or television monitor, the perception of the third dimension (depth) in the scene is diminished. The **goal** of 3D display technologies is to provide the viewer with a heightened perception of the depth in the scene. To achieve this perception of depth, a different view of the scene is provided to each eye of the viewer. Various methods have been adopted to display these two views. Some approaches require **glasses** to be worn by the viewer to direct the appropriate view to each

需求

eye while others don't require the viewer to wear glasses. However, these types of displays, which are known as **autostereoscopic** displays, require **multiple stereo views** of the scene to accommodate a range of **different viewing angles**. Stereoscopic video contains two separate views of a scene corresponding to the views required by the left and right eyes of the viewer. Multi-view video contains more than two views of a scene captured at different viewing angles and provides the different stereo views required for autostereoscopic displays.

应用

Currently, the major **applications** for stereoscopic video are 3D cinema and 3D television. The provision of stereo views gives the viewer a perception of the depth of the scene and this in turn provides the viewer with a more **immersive viewing experience** [1]. The **motivation** for providing the stereoscopic views in both 3D cinema and 3D TV is to make the experience more realistic for the viewer. Since humans naturally perceive the world in three dimensions, the provision of the perception of depth adds to the sense that the viewer is present in the scene. From the content creator's perspective, the more the 3D display techniques can add to the suspension of disbelief, the more **enjoyable** the viewing experience will be and ultimately the more incentive there will be for viewers to pay for that experience [2].

需求

With the advent of systems that are able to **acquire** and **display** multiple views of a scene comes the inherent requirement for **compressing** the large amounts of associated data for the purposes of **storage** and **transmission**. The coding algorithms developed for this compression task can be divided into two main approaches: those that compress the multiple views from each **camera** and those that compress some **combination** of the camera views and the depth information for the scene. The current status of standardization involves only the first approach with the **multi-view coding (MVC)** extension of the H.264/AVC standard supporting the prediction of a video frame from other views of the same scene. Support for coding **multi-view video with associated depth information** is currently being investigated for future standards.

2分类

当前

需求

包含

An inherent requirement for the development of multi-view video coding algorithms is the ability to **evaluate** the quality of the reconstructed views. For multi-view video there is an added requirement to evaluate the quality of the **complete 3D viewing experience**. These extra requirements involve evaluating the quality of **depth** provided by the video and identifying the artifacts associated with this type of display technology.

The remainder of this chapter contains a more detailed discussion on stereoscopic vision, the technology available for capturing and displaying multi-view video, the coding algorithms used for compressing the associated large amounts of data, and the perceptual evaluation of the resulting compressed video.

原因

Humans naturally perceive the world around them in three dimensions. This ability to perceive depth is mainly possible because of the **binocular vision** provided by two forward facing eyes. However there are many cues that the brain uses to estimate the distance to an object. These **depth cues** can be divided into three main categories: **proprioceptive**, **monoscopic**, and **stereoscopic** [3]. Only some of these cues require binocular vision.

### 5.04.2 Fundamentals of stereoscopic vision

描述

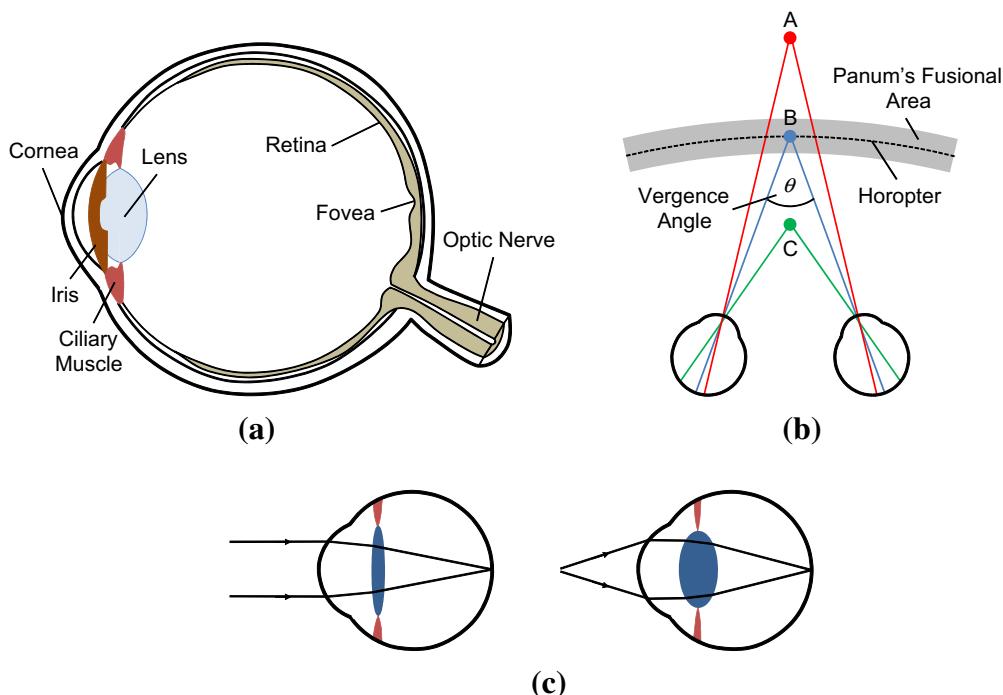
The term **proprioception** is used to describe how the brain senses the **spatial orientation** and **movement** of parts of the body using only stimuli from brain signals that control muscle movement. The brain

2要素

interprets the muscle movements in the eye required to view an object at a particular distance as a single object, and bring it into focus, as cues to the depth of the object. These two main cues which the brain interprets to estimate the depth of an object are known as **vergence** and **accommodation**.

The term **vergence** is used to describe the amount of **rotation** the eyes undergo to place the light rays emitted from an object at the same point on the **fovea** of both left and right eyes. A diagram of the anatomy of the eye is shown in [Figure 4.1a](#). As the object moves closer, the viewer's eyes must rotate inwards to keep the light rays from the object **centered** on the fovea as shown in [Figure 4.1b](#).

For any vergence angle there exists a theoretical **locus** of points where objects located at these points will be perceived as a single object as shown in [Figure 4.1b](#). This locus is known as the **horopter**. In practice however there is a **region** surrounding the horopter, known as **Panum's fusional area**, in which objects are perceived as a single object. If an object is located outside Panum's fusional area, light rays from the object will fall on different parts of the retina in each eye and it will be perceived as two objects. This double-vision effect is known as **diplopia**.



**FIGURE 4.1**

(a) Diagram of the anatomy of the human eye, (b) the vergence angle required to keep the light rays from an object centered on the fovea increases as the object moves closer to the viewer. (c) The shape of the lens in the eye is wider and thinner when focusing the light rays from a distant object on the back of the retina. To focus closer objects the lens shape is narrower and thicker.

The term **accommodation** is used to describe the change in focal length required by the lens in the front of the eye to keep the object of interest in focus. As the object moves closer to the viewer, ciliary muscles change the shape of the lens to decrease the focal length and keep the light rays from the object focused on the back of the retina, as shown in [Figure 4.1c](#).

The human brain has evolved to use the **muscle movements** that produce the correct vergence and accommodation for viewing a single focused object, as cues that indicate the distance of that object from the viewer. In stereoscopic displays, by providing the eyes with different views, the viewer naturally applies the correct amount of vergence to place the object at the fovea in each eye. The viewer's brain then interprets this vergence cue to perceive the depth of the object as being different from the depth of the display (the distance between the viewer and the screen). Unfortunately the accommodation required to place the left and right views in focus at the fovea must match the depth of the display not the perceived depth of the object. This vergence accommodation **conflict** is known to be the cause of some visual discomfort when viewing poorly presented 3D content [4,5].

冲突

The brain also uses cues from the scene being viewed as **indicators of the depth of objects**. Monoscopic depth cues are present in the view seen by a single eye. Stereoscopic depth cues are associated with the difference between the views seen by the left and right eye.

### 5.04.2.2 Monoscopic depth cues

**Relative size:** If a scene contains two objects that have a known size relative to each other, e.g., a person and a building, and the smaller object appears to be the same size as the larger object then the brain interprets this to mean that the person is closer than the building.

**Perspective and texture gradient:** If a scene contains a repetitive pattern, the pattern will appear smaller as the distance to the pattern increases. Similarly, if a scene contains parallel lines, e.g., railway tracks, the lines will appear closer together as the distance to them increases.

**Occlusion:** An object that is partially covering another object must be closer than the object it is covering.

**Shadows and specular reflections:** In most natural scenes there is only one light source such as the sun, the moon, or from overhead room lighting. The shadows cast by this one source of light provide cues to the 3D shape of an object. Similarly, the specular reflections of the light source from shiny surfaces also provide 3D shape cues.

**Motion parallax:** The term parallax refers to the relative position of an object in different views of the same scene. Motion parallax occurs when the viewer's position moves relative to the scene. After this change in viewer position, objects that are closer to the viewer will appear to have moved a greater distance than objects that are further away.

### 5.04.2.3 Stereoscopic depth cues

**Stereo parallax (Stereo disparity):** Stereo parallax refers to the change in position of objects between the left and right view because of the small difference in viewing position of the left and right eyes. As in motion parallax, objects that are closer will appear to move more between the left and right views than objects that are further away. Differences between the left and right views are called **retinal disparities** so stereo parallax is often referred to as stereo disparity. Since the difference in viewing position for the left and right eye is usually a horizontal shift, the main depth cue associated with stereo disparity is the change in the horizontal position of objects between the left and right views.

**Occlusion Revelations:** A closer object that is partially occluding a distant object will cover different parts of the distant object in the left and right views.

### 5.04.3 3D display technologies

In binocular vision the left and right eye see slightly different versions of the object. This effect was first reported by Sir Charles Wheatstone in 1838 [6]. The underlying requirement of 3D displays is to provide a means for the left and right eye to see the different views that would be seen if the object being displayed were at a particular distance from the viewer.

The techniques for displaying 3D images and video can be divided into the following broad categories: **direct-view**, **head-mounted**, and **volumetric** [7]. For the direct-view technologies the images are displayed on a planar surface such as the screen in a cinema or a television monitor. Alternatively, head-mounted technologies utilize a display system which is worn by the viewer and consists of goggles which allow the different stereo views to be displayed directly in front of the eyes of the viewer on special lenses. Volumetric displays form a visual representation of an object in three physical dimensions rather than the two dimensions of a screen or monitor. The most common approach in this category is the **swept volume display** where different views of an object are displayed on a rapidly spinning mirror or other display surface. This type of display will not be considered further in this chapter. For a comprehensive overview of the various volumetric display technologies the reader is referred to [8].

The **direct-view** technologies can be further divided into **stereoscopic** and **autostereoscopic** displays [9]. For **stereoscopic** displays the view for both the left and right eye is displayed **simultaneously** on the screen or monitor and the viewer is required to wear **special glasses** which allow only the appropriate view to be seen by each eye. **Autostereoscopic** display technologies provide a mechanism for the left and right views to be seen by the corresponding eye of the viewer without the need for special glasses. Direct view technologies are now in common usage and are considered in more detail in the following subsections.

#### 5.04.3.1 Stereoscopic displays

The types of glasses used in stereoscopic displays can also be divided into three main **types** depending on the **method** used to **filter** the correct view for each eye. The three types of filtering mechanisms use **color**, **polarization**, and **time** to separate the views for each eye.

The earliest stereoscopic displays implemented the **anaglyph** method where color is used to isolate the view for each eye [10, 11]. In this system, the stereo views are displayed on the screen or monitor using **complementary colors** (usually red and cyan or red and green) and the anaglyph glasses worn by the viewer filter out the unwanted view using different colored lenses for each eye. The main **limitations** of this approach are the **restriction of the colors** that can be seen in the image content and **crosstalk** between the views where each eye sees a portion of the view that is meant for the other eye. Many variations of this technique have been developed in recent times. One of the more advanced modifications of this approach involves the use of glasses with **complex color filters** which pass light through at three separate groups of **wavelengths** [12]. The three wavelength groups that are passed are different for each lens and complementary so that the range of colors seen by the viewer is almost identical to the full range of colors in the original content.

Since the 1950s **polarization** systems have been the dominant 3D display technology used in cinemas [13]. In this approach, the left and right views are projected onto the cinema screen through

实现

分类

原理

不足

polarization filters. The glasses worn by the viewer also contain polarization filters which only pass polarized light at the same orientation as the required view (e.g., horizontally polarized light for the left view and vertically polarized light for the right view). Standard white cinema screens tend to **scramble** the polarization of the projected light so screens coated with a silver metallic surface are used to maintain the polarization effect. The **requirement** for these silver screens and the polarizing filters for the projector and glasses increase the cost of polarization systems when compared to the simple complementary color anaglyph systems. However, the use of polarization display technology removes the **color limitations** that are inherent in the simple anaglyph methods.

Early polarization systems used linearly polarized light but the **disadvantage** of this approach is the **requirement** for the viewer to strictly maintain their eyes in a horizontal position. Any tilt in the viewer's head position allowed some light from the opposite polarization to be viewed producing **crosstalk** between the two views. This problem was alleviated by the introduction of systems which used **circularly polarized light** with opposite **handedness** for each view. A quarter waveplate can be used to convert circularly polarized light into linearly polarized light. The orientation of the waveplate and the handedness of the incident light determines the orientation of the linearly polarized light produced. A typical system uses left-hand circularly polarized light for the left view and the waveplate in the lens for the left eye produces linearly polarized light with an orientation of 45° relative to the top edge of the lens. Similarly the right view is projected using right-hand circularly polarized light and the right lens waveplate produces linearly polarized light with an orientation of 135°. Each lens also contains a polarization filter behind the quarter waveplate to only allow light at the correct orientation to pass through to the eye of the viewer. The advantage of this system is that the handedness of the incoming circularly polarized light is not affected by the orientation of the viewer's head. This allows the viewer to perceive an acceptable 3D effect for a much wider range of head positions, which improves the comfort of the viewing experience.

The third type of stereoscopic display technology involves the use of **active shutters** in the lenses of glasses worn by the viewer [7,9]. In this approach the left and right views are displayed at **alternate times** and the shutters in the lenses switch from being **opaque** to **transparent** to either transmit the desired view or block the view for the other eye. The **frequency** at which the views alternate and the lenses switch must be such that the viewer does not perceive any noticeable flicker in the displayed images. Generally an overall frame rate of **120 Hz**, which corresponds to **60 Hz** for each view, is considered **acceptable**. The active shutters required for this system are generally composed of glass with a **liquid crystal layer** embedded. The liquid crystal layer becomes opaque when a voltage is applied across it and is transparent when the voltage is removed. In active shutter systems, the shutters in the lens must be **synchronized** to the display of each view. This is achieved using a wired or wireless communications link between the display and the glasses. Wireless links are now more common and use optical or radio frequency (RF) signals. The main **disadvantage** of this type of 3D display is the **expense** and **complexity** of the electronics required for the communications link to the display and the control signals for the liquid crystal layer. Although this type of display has been used in cinemas, the main applications for the active shutter type of displays are for 3D televisions and gaming consoles.

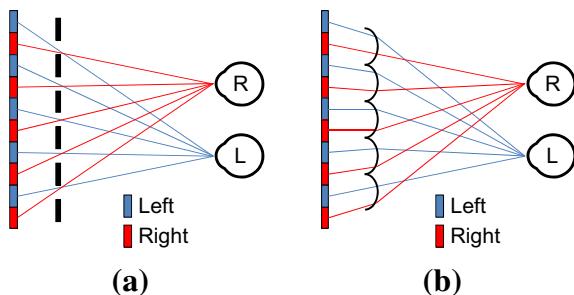
### 5.04.3.2 Autostereoscopic displays

Autostereoscopic displays can be divided into several categories according to the **number of pairs of stereo views provided**.

同步

不足

类化

**FIGURE 4.2**

The principle of two-view autostereoscopic displays using (a) parallax barriers and (b) lenticular lenses.

#### **5.04.3.2.1 Two-view systems**

The first type of autostereoscopic display provides a single stereo pair of views. The methods for providing a separate view for each eye without the use of special glasses vary according to the particular design of the system. The main methods for separating the views use either **parallax barriers** or **lenticular lenses** in a thin layer on the front of the display.

In displays that use a parallax barrier, the left and right views are displayed in alternating vertical columns. A parallax barrier, which consists of vertical apertures spaced appropriately, is placed in front of the display and allows the two views to be seen by different eyes if the viewer is situated within the corresponding viewing zone as shown in Figure 4.2a.

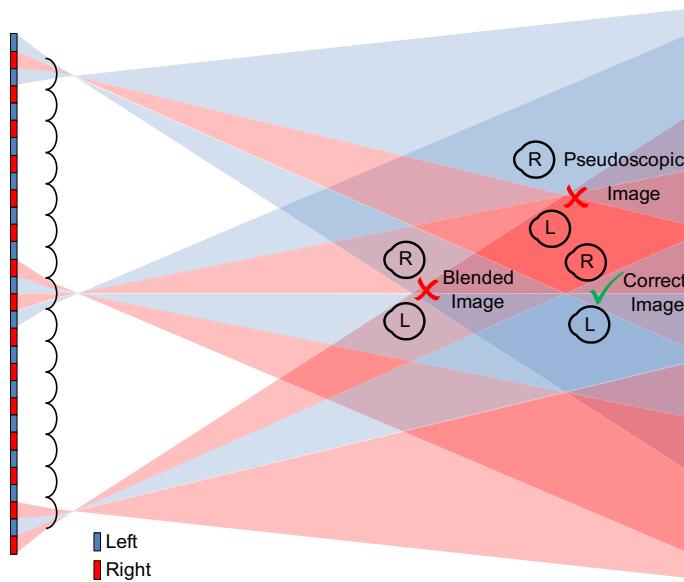
A lenticular lens consists of a series of semi-cylindrical lenses arranged side-by-side. As with parallax barrier displays the left and right views are displayed in alternating columns, with a lens placed in front of each pair of columns. The cylindrical lens will diffract light from the underlying display so that the left and right views appear in alternating regions of the viewing zone as shown in Figure 4.2b.

The main **problem** with two-view systems occurs when the viewer moves out of the ideal **viewing zone**. When the viewer's eyes are placed so that the left eye sees the right view and the right eye sees the left view a pseudoscopic effect is produced and depth perception is **reversed**. Similarly if the viewing distance is different from the ideal distance the viewer will see a blend of both views in each eye as shown in Figure 4.3.

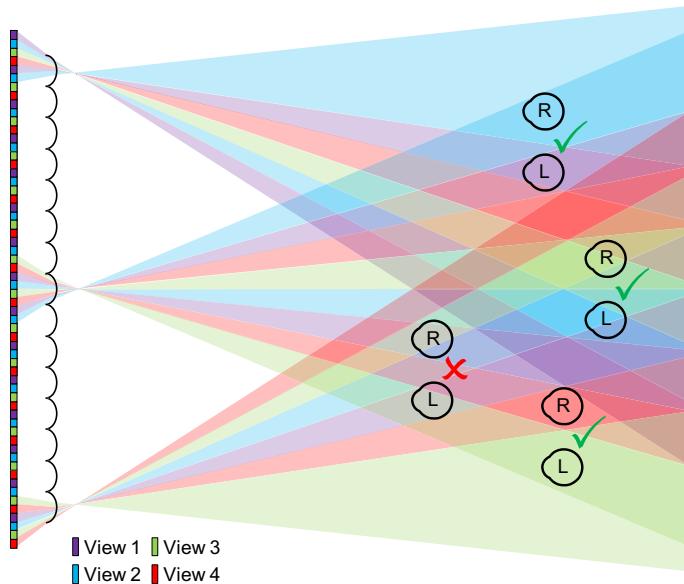
To overcome the problem of the small viewing zone in static two-view autostereoscopic displays, an **eye tracking system** can be used to adjust the position of the correct viewing zones to match the position of the viewer's eyes. Various methods have been developed to adjust the position of the stereo views for both parallax barrier and lenticular lens type displays.

#### **5.04.3.2.2 Multi-view systems**

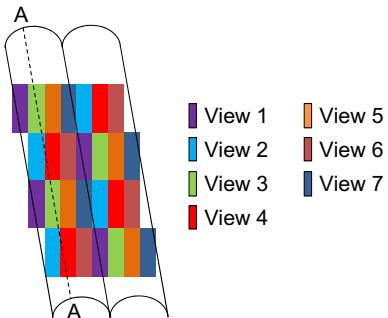
In multi-view autostereoscopic displays, multiple adjacent views of the same 3D scene are provided in the **viewing zone**, as shown in Figure 4.4 for a display that provides four views. For these displays, as the viewer's eyes move to the left and right, the view changes to provide a different perspective of the same scene. For example, for the display shown in Figure 4.4, as the viewer's head moves from left to right the view pairs seen by the left and right eye respectively will change from 1, 2 to 2, 3 and then to 3, 4.

**FIGURE 4.3**

The multiple viewing zones produced by a two-view autostereoscopic display.

**FIGURE 4.4**

Viewing zones for a four-view autostereoscopic display.

**FIGURE 4.5**

The use of a slanted lenticular lens array allows the resolution reduction for each view to be distributed between the horizontal and vertical directions.

In contrast, for a two-view display the view pairs seen by the viewer for the same head movement would be L, R followed by R, L and then L, R again. A multi-view display allows a **wider viewing zone** and provides a limited amount of motion parallax. Although the number of views provided in multi-view displays is considered to be too small to provide a continuous motion parallax effect [14].

A **limiting factor** for this approach is the **resolution** of the display that provides the multiple views. Since the separate views are displayed in adjacent columns, as the number of views increases, the horizontal resolution of each view decreases. To overcome this limitation, multi-view displays have been developed which use **slanted lenticular lenses** in combination with a pixel-to-view mapping that spans two rows of the display as shown in [Figure 4.5](#) for a seven-view display [14]. In the slanted view approach, as the viewers head moves from left to right, the views gradually transition from view 1 to 7 and, since adjacent pixel mappings alternate the rows used for each view, the viewer does not see a noticeable vertical shift between views. The pixels contributing to the view at a particular viewing angle will be those pixels at the same horizontal position on each lens. Since the lenses are slanted, pixels corresponding to adjacent camera views will contribute to the view seen at each viewing angle. For example, for the view angle corresponding to the horizontal position shown by line A-A, pixels belonging to camera view 3 will provide the most light intensity, with lesser contributions from camera views 2 and 4. In this way, a display with a slanted lenticular lens provides gradual transitions between multiple views and distributes the resolution reduction for each view between the horizontal and vertical directions.

#### **5.04.3.2.3 Super multi-view systems**

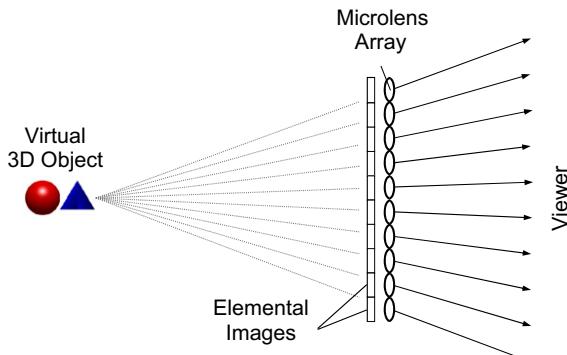
The term super multi-view display is used to describe a display where the number of views is **sufficient** to produce the **effect of continuous motion parallax**. It has been shown that, if the **width** of each view is small enough to allow two views to enter the pupil, the eye is able to focus at depths other than the screen depth [15]. Because the eyes are now able to focus on the virtual depth of the object, the viewer is able to perceive a more natural 3D image without any accommodation convergence conflict. Practical limitations however limit super multi-view displays to provide views with wider individual views than this.

Different approaches have been used to build super multi-view displays. One approach uses a rapidly changing vertical display behind a **vertical shutter system**. By synchronizing the shutters with the display

system each view is provided for a small time interval. Systems using this approach have been developed that provide up to 28 views [16].

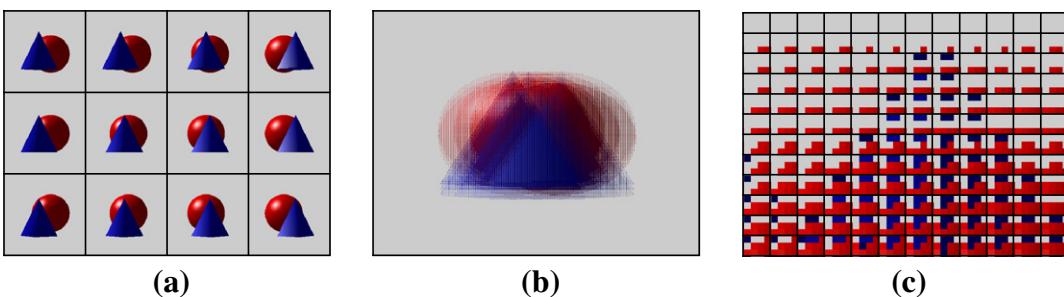
**Integral imaging** is an alternative approach to providing super multi-view displays [17]. Integral imaging is basically an **extension** of the lenticular lens type systems used for multi-view displays. This approach involves the use of a microlens array in front of a high resolution display as shown in [Figure 4.6](#). The area of the display behind each lens is called an elemental image and provides the light intensities corresponding to multiple viewing angles in both the horizontal and vertical directions. [Figure 4.7](#) shows the individual views and the displayed elemental images for a simplified integral imaging system with only 12 views.

The number of views provided by integral imaging systems is currently limited by the resolution of the display technology used to produce the elemental images. For LCD approaches the latest developments to overcome this problem, include using rectangular elemental images and slanted microlens arrays to



**FIGURE 4.6**

An integral imaging system consists of a microlens array in front of a display containing multiple elemental images.



**FIGURE 4.7**

(a) The multiple views for an integral imaging system with 12 views, (b) the combined display showing multiple elemental images, (c) a close-up view of the  $3 \times 4$  pixel elemental images.

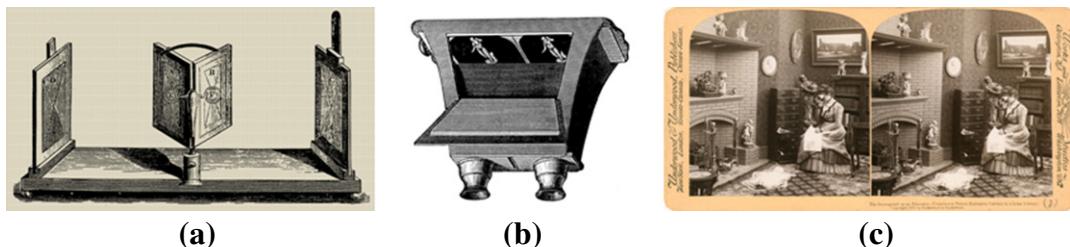
provide a larger horizontal viewing angle [18]. An alternative approach for producing the elemental images is to use a projector array. A prototype system that provides 60 views was recently demonstrated [19]. This approach has the potential to allow more views since the number of views in each elemental image is not limited by the pixel resolution of an LCD screen.

#### 5.04.4 Applications of stereoscopic and multi-view video

The applications of stereoscopic and multi-view video fall into two main categories: niche applications where depth perception is essential for the success of the application, and widely adopted consumer applications where depth perception is not essential but adds to the user experience of the entertainment service when compared to its 2D equivalent. Visualization of complex 3D structures in scientific and medical applications and remote control of robotic vehicles are examples of applications in the first category. Applications that fall into the second category are 3D cinema, 3D television, and 3D gaming. Of these commercial applications, 3D cinema and 3D television are currently receiving the most attention from the consumer devices and entertainment industries.

##### 5.04.4.1 3D cinema

The first device that could be used for viewing a pair of stereo images as a 3D object was invented by Sir Charles Wheatstone in 1833. This device had a pair of mirrors angled at 45° which reflected light from left and right images located at each side onto the corresponding eye of the viewer at the front of the device. An illustration of Wheatstone's device is shown in Figure 4.8a. Soon after the invention of photography in the 1840s, the Scottish scientist Sir David Brewster developed a device called a stereoscope which used a pair of lenses to provide a 3D view from a pair of stereo photographs [20]. Figure 4.8b and c show an illustration of Brewster's stereoscope and an example of a card holding stereo photographs. This device became the template for many other similar devices used for viewing stereo photography. The View-Master stereoscope is a modern implementation of Brewster's original concept that uses 35 mm color film to display the stereo image for each eye. Queen Victoria initiated a surge in



**FIGURE 4.8**

- (a) The first known device for viewing a pair of images as a 3D object, invented by Sir Charles Wheatstone, (b) the Brewster stereoscope, (c) an example of a card containing left and right images for viewing in an early stereoscope.

the popularity of stereo photography in the late 1800s. She became an avid user of stereo photography after viewing 3D images through a stereoscope at the World's Fair in London in 1851.

The popularity of stereo photography began to decline with the advent of cinematography in the early 1900s. As with photography, the development of 3D cinema began soon after its 2D counterpart [21]. In the late 1890s William Friese-Green patented a 3D movie approach that involved projecting left and right views onto a screen and viewing them through a stereoscope. However, due to the complex projection and viewing mechanics required, this method was not practical for large scale use in theaters. The first test of the anaglyph viewing process for a theatrical audience occurred in 1915 at the Astor Theater in New York and in 1922 the first paying audience watched the film "The Power of Love" using red-green anaglyph glasses at the Ambassador Hotel Theater in Los Angeles. The anaglyph process continued to be used until the late 1930s when the polarized light approach was introduced.

Military applications of stereo photography were considered a priority during the second world war so it wasn't until the 1950s that 3D cinema again began to be considered by movie producers. The first full-color film using the polarization display approach, "Bwana Devil," was released in 1952 and sparked a resurgence in the popularity of 3D cinema. This popularity waned during the 1960s and 1970s with films being released sporadically. Then, in the mid 1980s, IMAX began producing documentaries in 3D for display in theaters with a very wide field of view which provided a more comfortable viewing experience than previous 35 mm approaches. The latest mainstream resurgence of 3D cinema started in the early 2000s with the development of high definition digital video cameras and the computer graphics technology that allowed 3D versions of fully animated 2D movies to be easily produced.

#### 5.04.4.2 3D television

Soon after the introduction of broadcast television in 1924, its pioneer, Scottish inventor John Logie Baird, also demonstrated a stereoscopic television system at his company's premises in London. Baird later proposed a variety of 3D television systems using electromechanical and cathode ray tube technologies. However, due to their technical complexity Logie Baird's 3D television systems were not adopted and broadcast 3D TV was not introduced until much later in the 20th century.

In the United Kingdom, the first 3D broadcast was an episode of "Tomorrow's World," a weekly science magazine program, which screened in February 1982. The broadcast included excerpts of test footage shot in the Netherlands. Viewers watched the program using red/green anaglyph glasses that were given away free with copies of a TV guide. For this first screening, the 3D sections of the program were shown in monochrome but in December 1982 the broadcast was repeated in color using red/blue glasses. In the United States in May 1997, parts of two episodes of "3rd Rock from the Sun" were shown in 3D. Viewers were signaled to put on their 3D glasses using "3D ON" and "3D OFF" icons in the corner of the display.

With the development of passive polarized and active shutter television displays in the mid 2000s the quality of the 3D TV viewing experienced increased markedly when compared to the early anaglyph broadcasts. In 2010, the first live sports event was broadcast in 3D when a football match between Manchester United and Arsenal was broadcast to a public audience in several selected venues. Also in 2010, the French Open tennis tournament in Paris was broadcast live via dedicated ADSL and optical fiber channels to viewers throughout France.

Since these initial trials, many major sporting events have been broadcast in 3D including the 2010 FIFA world cup, the 2010 PGA golf tournament, and the 2011 Rugby World cup, as well as selected NASCAR races and NFL, Rugby League, and Australian Rules games. Other special events that have been broadcast in 3D include the opening and closing ceremonies of the 2012 Olympic games, the inauguration of Philippine's president Noynoy Aquino and the 2010 tour of Canada by Queen Elizabeth II of Britain. In January 2011, "Tokyo Control" became the first Japanese television series to be broadcast in 3D. As well as these broadcasts of special events in 3D, there are now over 30 TV channels that are exclusively transmitting 3D content in more than 15 different countries around the world.

The data format currently used to broadcast 3D TV is known as Frame Compatible Stereo [22]. In this format, the original, full resolution, left, and right views are down-sampled and multiplexed into a single frame. The different options for this resolution reduction and multiplexing are: Side-by-Side, where the horizontal resolution is reduced, Line-by-Line and Top-Bottom, where the vertical resolution is reduced, and Checkerboard, where both the horizontal and vertical resolutions are reduced [23]. This format is compatible with existing digital broadcast television standards but has the obvious disadvantage of the required reduction in display resolution.

Free-viewpoint TV (FTV) is a logical future extension of 3D TV where the viewer is able to interactively select the viewpoint from a large range of near-continuous viewing angles [24–26]. FTV will provide the user with a more realistic and immersive viewing experience however the super multi-view display technologies required for these services are not currently commercially available [27].

## 5.04.5 Compression of stereoscopic and multi-view video

With the introduction of video systems that are able to acquire and display multiple views of a scene, there is now a requirement for compressing the large amounts of associated video data for the purposes of storage and transmission. The following section contains a brief introduction to the underlying concepts and basic strategies used in video coding.

### 5.04.5.1 Introduction to video coding

The compression of video sequences involves the use of a variety of lossless and lossy data compression approaches. By using these approaches the current video coding standards are able to provide **compression ratios** of approximately **80:1** for typical video content. A brief description of the main data compression concepts and techniques that are used in video coding is provided in the remainder of this section.

#### 5.04.5.1.1 Information

In communications theory, the **information** contained in a particular message is related to the probability that the message will be transmitted. Mathematically, the information content,  $I$ , of a message is defined by the equation:

$$I = \log_2 \left( \frac{1}{P} \right) \text{ bits}, \quad (4.1)$$

where  $P$  is the **probability** that the message will be transmitted. Hence, the higher the probability that the message will be sent, the less information the message contains. If this equation is used, the **information**

in a message is measured in bits. Once the **information content** of each message is known, the **average information content** of the source, called the **entropy**, can be calculated. The entropy,  $H$ , of a source is given by the equation

$$H = \sum_{i=1}^M P_i \log_2 \left( \frac{1}{P_i} \right) \text{ bits/message}, \quad (4.2)$$

where  $M$  is the number of different messages produced by the source and  $P_i$  is the probability of transmitting the  $i$ th message.

#### 5.04.5.1.2 Entropy coding

The data compression technique known as entropy coding makes use of the different information content of each message by using **variable length patterns** of bits (called **codewords**) to represent each message. As a general rule, if smaller codewords are allocated to the messages which are more likely to be transmitted, the average number of bits required to be transmitted is less than if all messages were allocated equal length codewords. Two common examples of entropy coding techniques are **Huffman coding** and **arithmetic coding**.

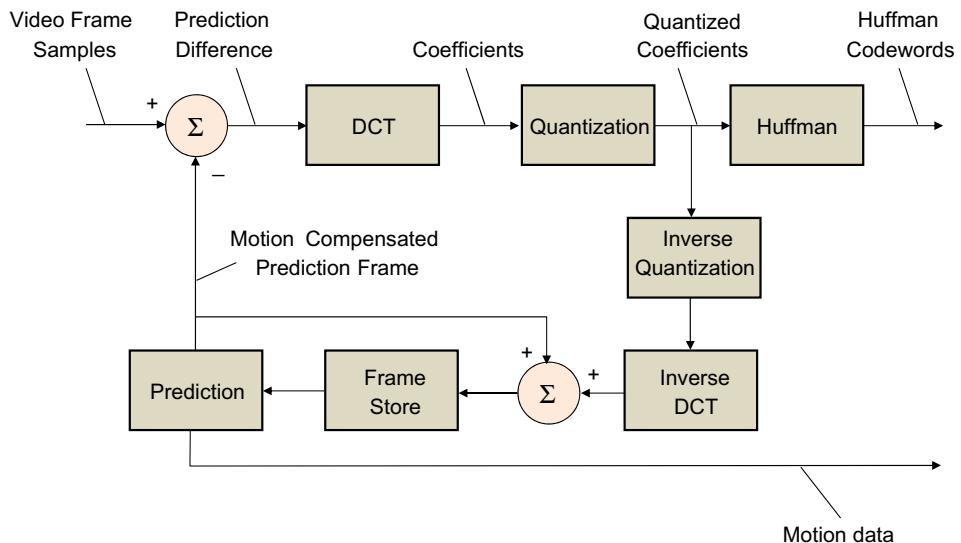
#### 5.04.5.1.3 Correlation and entropy reduction

There is often a large amount of **correlation** between **neighboring samples** of a typical image or video signal. This correlation can be exploited to reduce the entropy of the information source. The two main techniques for reducing the entropy of the source are **predictive coding** and **transform coding**.

In predictive coding a part of the original signal which has already been sent to the decoder is used to predict the current sample value. This prediction is then subtracted from the current sample and only the **difference** between the prediction and the original sample value is **transmitted**. At the decoder, the same prediction can be generated and the difference value is added to the prediction to obtain the original signal value. If there is a large amount of **correlation** between successive samples, the **entropy** of the difference values will be much less than the entropy of the original sample values. Note that predictive coding is a lossless compression technique since the original sampled signal can be perfectly reconstructed at the decoder.

In transform coding, the original sample values are transformed into a different domain by finding the cross product of the input samples with a set of orthogonal **basis functions**. These basis functions usually span a block of pixels with typical sizes of  $16 \times 16$  down to  $4 \times 4$  pixels. The basis functions can be sampled sinusoids of increasing frequency and the resulting coefficients indicate the frequency content of the original waveform. At the decoder, each basis function is multiplied by its corresponding coefficient and the resulting scaled basis functions are added together to produce the original samples.

The transform coefficients for each block of the video frame are converted to integer values using quantization. The quantizer step-size is defined as the smallest difference between two input coefficient values that will produce two different output integer values. The process of quantization does of course introduce distortion which is known as quantization noise. If the original samples are highly correlated, then many of the high frequency coefficients will have values close to zero. If the set of coefficients is quantized, many of the high frequency coefficients will be set to zero. The entropy of the resulting set of quantized coefficients is typically much smaller than the entropy of the original samples. Since the original samples can be represented using the low frequency coefficients, often only a small number of

**FIGURE 4.9**

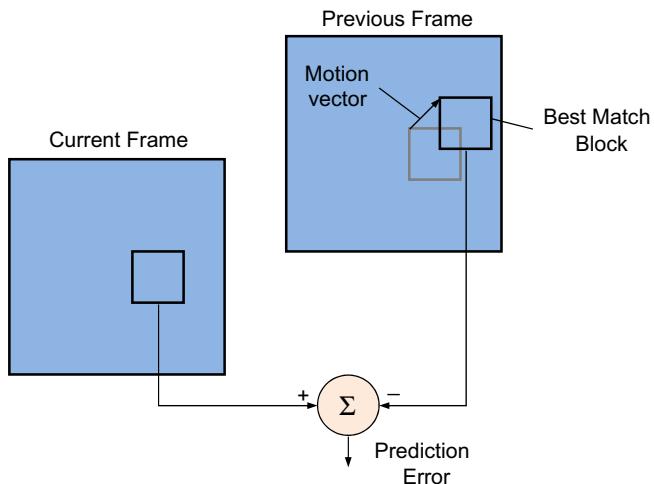
The basic block diagram of a block-based motion-compensated video coder.

quantized low frequency coefficients are needed to produce a good quality reproduction of the original image. Note, however, that transform coding is a lossy compression technique and there will always be some degradation of the quality of the reconstructed signal caused by the quantization of the transform coefficients. Using a larger step-size in the quantization process will result in less bits being transmitted and a lower quality reconstructed signal.

The general **block diagram** of a video coding algorithm is shown in [Figure 4.9](#). The main compression strategies used in video coding are entropy reduction, using prediction and the quantization of transform coefficients, combined with efficient transmission using entropy coding. Since there is significant correlation between the red, green, and blue components of color video frames, these components are transformed into a **luminance** and **chrominance** representation. The two chrominance components contain considerably less information than the luminance component so the spatial resolution of these components is often reduced prior to compression.

#### **5.04.5.1.4 Inter-frame prediction**

In video coding, previously coded frames are used to predict the current frame. For typical video sequences, there is significant correlation between one frame and the next. So, instead of coding the current frame directly, the difference between the current frame and a previously coded frame is coded. This type of coding is called **inter-frame predictive coding**. Note that the decoded version of the previously coded frame must be used in the prediction since this is the version of the frame available at the decoder. In inter-frame coding, frame  $n - 1$  in the sequence will typically be a good predictor for frame  $n$ . However there will be some difference between these frames due to the motion of objects depicted in the video scene. So, in order to improve the prediction, motion compensation is used. For each block

**FIGURE 4.10**

The general approach used in block-based motion compensation.

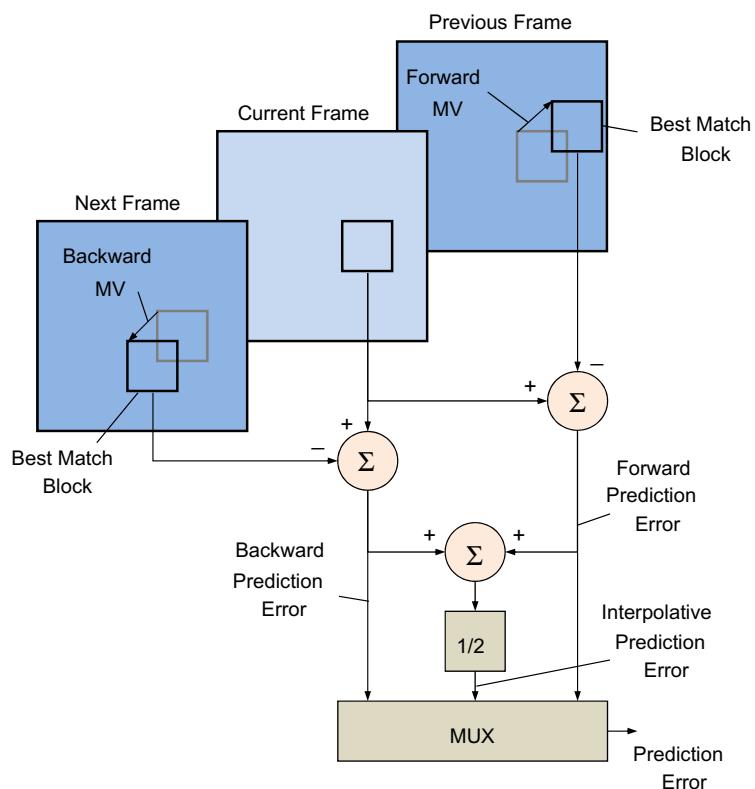
in the current picture, motion estimation is used to find the block in the previous frame which best matches the current block, as shown in [Figure 4.10](#). The vector which describes the translation from the position of the current block to the position of the best match block is called the motion vector (MV). The current block and the best match block are subtracted to form the motion-compensated difference block which is then coded. The motion vectors must also be entropy coded and sent to the decoder.

If frame  $n - 1$  is used to form the motion-compensated prediction for frame  $n$ , this is known as forward motion compensation. If frame  $n + 1$  is used to form the motion-compensated prediction for frame  $n$ , this is known as backward motion compensation. Note that for backward motion compensation frame  $n + 1$  must have been coded before frame  $n$  which will require some reordering and buffering of the input video frames. If both forward and backward motion compensation is performed for the current frame, this is known as bi-directional motion compensation. As shown in [Figure 4.11](#), for bi-directional motion compensation, the motion-compensated difference block can be formed using either the forward best match block or the backward best match block or the average of the two.

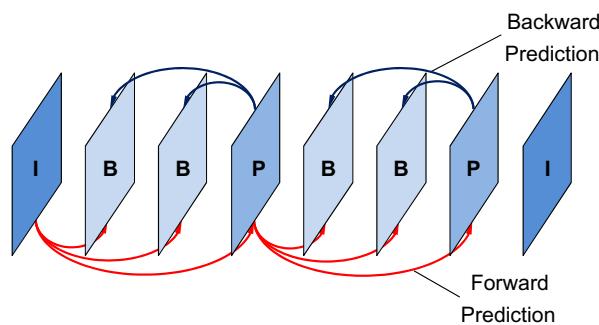
The frames of the video are classified into three types according to the way they are coded. Intra (I) frames are coded directly without any prediction from other frames. Predicted (P) frames are coded using forward motion compensation from a previously coded I or P frame. Bi-directional (B) frames are coded using bi-directional motion compensation from two previously coded I or P frames. Frames are usually coded in groups of I, P, and B frames with each group containing at least one I frame. A typical group of coded frames is shown in [Figure 4.12](#) with arrows showing the frames used to form the motion-compensated prediction of each B and P frame.

#### **5.04.5.1.5 Encoder optimization**

In the latest commercially available video coding standard, the encoder can choose between one of eight different block sizes when forming the motion-compensated prediction. Each  $16 \times 16$  pixel block

**FIGURE 4.11**

The general approach used in bi-directional block-based motion compensation.

**FIGURE 4.12**

The frame types in a typical group of coded frames when using bi-directional motion compensation.

may be divided into partitions in four ways: one  $16 \times 16$  partition, two  $8 \times 16$  partitions, two  $16 \times 8$  partitions, or four  $8 \times 8$  partitions. If the  $8 \times 8$  mode is chosen, each of the four  $8 \times 8$  sub-blocks may be divided into sub-block partitions or sub-partitions in four ways: one  $8 \times 8$  partition, two  $4 \times 8$  partitions, two  $8 \times 4$  partitions, or four  $4 \times 4$  partitions. A separate motion vector must be transmitted for each partition or sub-partition. The motion vectors can be specified to an accuracy of one quarter of a pixel for luminance blocks and one eighth of a pixel for chrominance blocks. The partitioning mode choice for each block and sub-block must also be transmitted.

The choice of block-size is a trade-off between reduced energy in the motion-compensated residual for smaller block sizes and the increased number of bits required to code the extra motion vectors and mode choice data. Typically larger block-sizes are best suited to large areas with homogeneous motion. In a rate-distortion optimized video encoder, the inter-mode choice is performed by selecting the mode which minimizes a Lagrangian cost function given by:

$$J_{\text{mode}} = D_{\text{mode}} + \lambda_{\text{mode}} R_{\text{mode}}, \quad (4.3)$$

where  $D_{\text{mode}}$  is the sum-of-squared difference between the original and reconstructed blocks or partitions,  $R_{\text{mode}}$  is the bit-rate required to transmit the motion vectors, transform coefficients of the prediction residual and block type information, and  $\lambda_{\text{mode}}$  is a Lagrangian multiplier. The choice of motion vector for a particular partition is also performed using a Lagrangian cost function given by:

$$J_{\text{motion}} = D_{\text{motion}} + \lambda_{\text{motion}} R_{\text{motion}}, \quad (4.4)$$

where  $D_{\text{motion}}$  is the sum-of-absolute difference between the current partition and the partition in the reference frame indicated by the candidate motion vector,  $R_{\text{motion}}$  is the bit-rate required to transmit the motion vectors, and  $\lambda_{\text{motion}}$  is a Lagrangian multiplier. Fortunately it has been shown that the optimal values for the Lagrangian multipliers in the mode choice equations are related to the level of quantization used to compress the transform coefficients of the prediction residual. Consequently, it is recommended that the Lagrangian multipliers for the rate-distortion optimizations should be calculated using the equations:

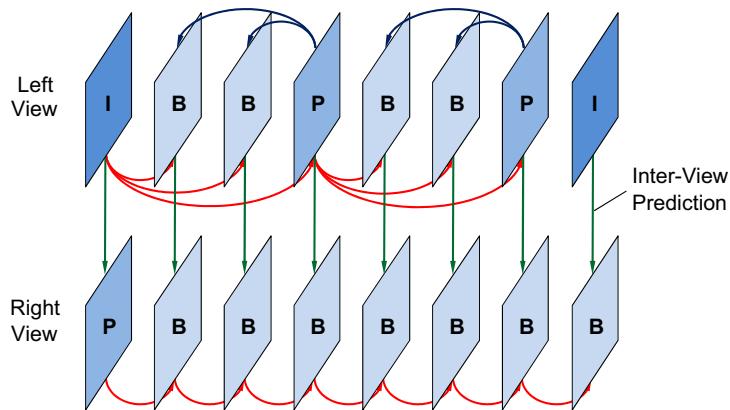
$$\lambda_{\text{mode}} = 0.85 \left( 2^{\text{QP}/3} \right), \quad (4.5)$$

$$\lambda_{\text{motion}} = \sqrt{\lambda_{\text{mode}}}, \quad (4.6)$$

where QP is the quantization parameter that specifies the step-size used to quantize the transform coefficients. It should be noted that in the next generation video coding standard, HEVC, an even larger choice of block sizes and partitions is available. For further details the reader is referred to [28].

### 5.04.5.2 Multi-view video coding using inter-view prediction

For video with multiple views, the correlation between views can also be used to reduce the entropy of the video sequence. The multi-view video coding (MVC) extension of the H.264/AVC standard adopts this approach [29,30]. Here, as well as temporal prediction, best match blocks for the current block are found in the same frame from other views as shown in Figure 4.13 for a sequence containing two views. The prediction is performed in a similar manner to motion compensation between temporally adjacent frames of the same view. The main difference between content in the two views will be a horizontal shift

**FIGURE 4.13**

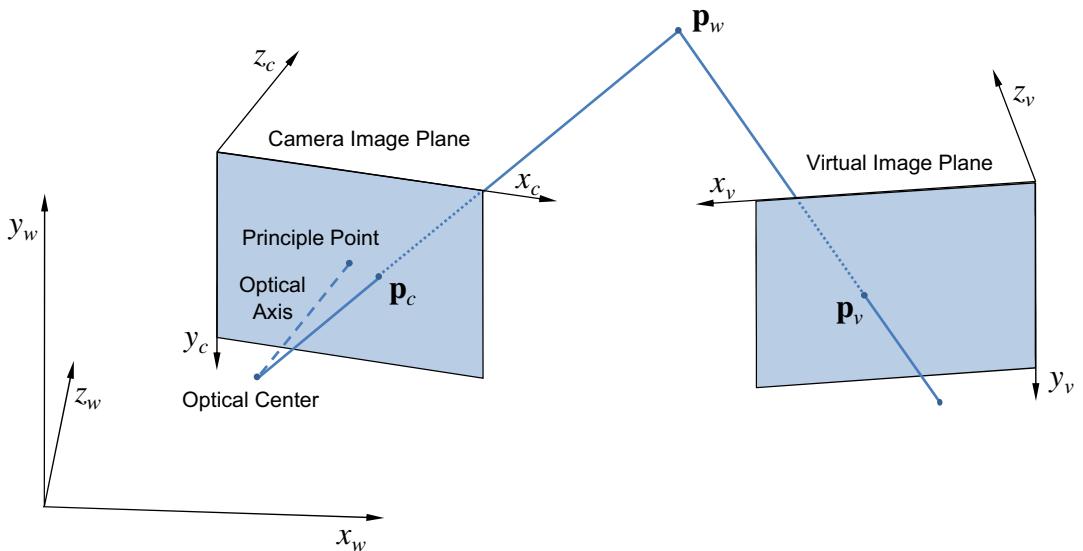
The frame types in a typical group of coded stereo frames when using bi-directional motion compensation and inter-view prediction.

between views due to the disparity caused by differing viewing angles. The position of the best match block in another view is transmitted to the decoder via a disparity vector. Although this prediction could be performed in both directions, in the MVC standard only one direction is allowed. This provides one view that can be decoded independently for compatibility with H.264/AVC decoders that only support a single view.

There are some **limitations** with the inter-view prediction approach provided in the MVC standard. The algorithms originally developed for motion compensation are applied unmodified for disparity compensation between views. Although motion between temporally adjacent frames and disparity between views both cause a change in the position of objects, the statistics of these positional changes are not the same. The entropy coding of motion vectors is based on the assumption that zero motion between frames has the highest probability whereas zero disparity between views typically occurs for only 20% of the frame. The disparity between views is determined by the setting of the camera array, such as the distance between cameras and the difference in viewing angle between views, as well as the depth of the objects in the scene. Coding gains of up to 40% have been achieved for MVC when compared to the separate coding of each view. The bit-rate produced by the MVC coder also increases linearly with the number of views that are coded [31].

The views required for different display technologies vary considerably. The number of views and the view angles provided must match the view angles expected by the display. If the content transmitted to these displays is coded using the MVC approach, a large number of views (up to 50) is required in order to provide the views required for each specific 3D display device. The bit-rate required to transmit this large number of views is prohibitive given the linear relationship between the bit-rate and number of views to be coded.

An alternative approach to transmitting each view with inter-view prediction is to transmit a depth-based representation of the 3D objects in the scene. This representation includes a subset of the views required and information describing the depth of the objects in the scene. When the depth information

**FIGURE 4.14**

The coordinate systems and pinhole camera models used in virtual view synthesis.

is available to the decoder virtual views can be synthesized at the decoder using a technique known as depth image-based rendering (DIBR) [32].

#### 5.04.5.3 View synthesis using depth image-based rendering

The procedure for generating a virtual view from a camera view and depth information relies on the following geometrical principles [33]. Consider the pinhole model for a camera shown in Figure 4.14. In the most general case the world coordinate system is different to the camera coordinate system. Now consider a point in 3D space with a position in the world coordinate system of  $\mathbf{p}_w = [x_w \ y_w \ z_w]^T$ . The position of this point in the camera coordinate system is given by  $\mathbf{p}_c = [x_c \ y_c \ 1]^T$  in homogeneous coordinate notation. The relationship between the position of the point in the world and camera coordinate systems is given by:

$$z_c \mathbf{p}_c = \mathbf{K}_c (\mathbf{R}_c \mathbf{p}_w + \mathbf{t}_c), \quad (4.7)$$

where  $\mathbf{R}_c$  is a  $3 \times 3$  **rotation** matrix that rotates the axes in the world coordinate system so that they are parallel with the axes in the camera coordinate system,  $\mathbf{t}_c$  is a  $3 \times 1$  translation vector that translates the origin of the world coordinate system to the origin of the camera coordinate system, and  $\mathbf{K}_c$  is a  $3 \times 3$  upper triangular matrix that specifies the intrinsic parameters of the camera. In the simplest case  $\mathbf{K}_c$  is given by:

$$\mathbf{K}_c = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.8)$$

where  $f_x$  and  $f_y$  are the horizontal and vertical focal lengths of the camera respectively. This simple form of the intrinsic parameter matrix can be used if the origin of the camera coordinate system is assumed to be located at the principle point (the intersection of the optical axis and the image plane), the aspect ratio of the pixels is 1:1 and no shearing (sometimes called skewing) occurred during the image capture process.  $z_c$  is the component, in the  $z$  direction of the camera coordinate system, of the distance from the point in 3D space to the optical center of the camera.

Now consider the case when a view is to be synthesized that corresponds to a different virtual image plane. The relationship between the position of a point in the world coordinate system and the virtual camera coordinate system is given by:

$$z_v \mathbf{p}_v = \mathbf{K}_v (\mathbf{R}_v \mathbf{p}_w + \mathbf{t}_v), \quad (4.9)$$

where  $\mathbf{p}_v = [x_v \ y_v \ 1]^T$  is the position of the point in the virtual image plane expressed in homogeneous coordinate notation and the other terms are defined as for the real camera coordinate system. To map from the real camera system to the virtual camera system, each pixel position is projected into the real world system using the depth information for each pixel as follows:

$$\mathbf{p}_w = \mathbf{R}_c^{-1} (z_c \mathbf{K}_c^{-1} \mathbf{p}_c - \mathbf{t}_c). \quad (4.10)$$

The position of the pixel in the virtual view can then be determined by substituting this equation into the equation for the virtual pinhole camera as follows:

$$\mathbf{p}_v = \frac{1}{z_v} \mathbf{K}_v (\mathbf{R}_v \mathbf{R}_c^{-1} (z_c \mathbf{K}_c^{-1} \mathbf{p}_c - \mathbf{t}_c) + \mathbf{t}_v). \quad (4.11)$$

This equation is quite complex and can provide inaccurate results for small errors in the intrinsic camera parameters or the depth information. Fortunately the calculations can be simplified greatly by choosing a specific camera arrangement and an appropriate coordinate system. First, the camera system is arranged in a parallel configuration where the image plane of the virtual view is placed on the horizontal axis of the camera view and the optical axis of the virtual view is parallel to that of the camera view. This arrangement results in only a horizontal shift between the camera image plane and the virtual image plane. Second, the world coordinate system is assumed to be the camera coordinate system. These simplifying conditions mean that the following assumptions hold:  $\mathbf{K}_v = \mathbf{K}_c$ ,  $z_v = z_c$ ,  $\mathbf{R}_c$  and  $\mathbf{R}_v$  are equal to the  $3 \times 3$  identity matrix,  $\mathbf{t}_c = [0 \ 0 \ 0]^T$  and  $\mathbf{t}_v = [t_x \ 0 \ 0]^T$  where  $t_x$  is the horizontal baseline shift between the real camera view and the virtual camera view. After applying these simplifications, the position of each pixel in the virtual view is given by:

$$\mathbf{p}_v = \mathbf{p}_c + \frac{1}{z_v} \mathbf{K}_v \mathbf{t}_v \quad (4.12)$$

and, since  $\mathbf{t}_v$  contains only a horizontal non-zero component, this equation can be further simplified to give:

$$x_v = x_c + \frac{f_x t_x}{z_v}. \quad (4.13)$$

Rearranging this equation also provides the horizontal disparity  $d_x$  between the position of each pixel in the real camera view and its new position in the virtual camera view:

$$d_x = x_v - x_c = \frac{f_x t_x}{z_v}. \quad (4.14)$$

Due to the arrangement of the camera system,  $y_v = y_c$  and the vertical disparity is equal to zero.

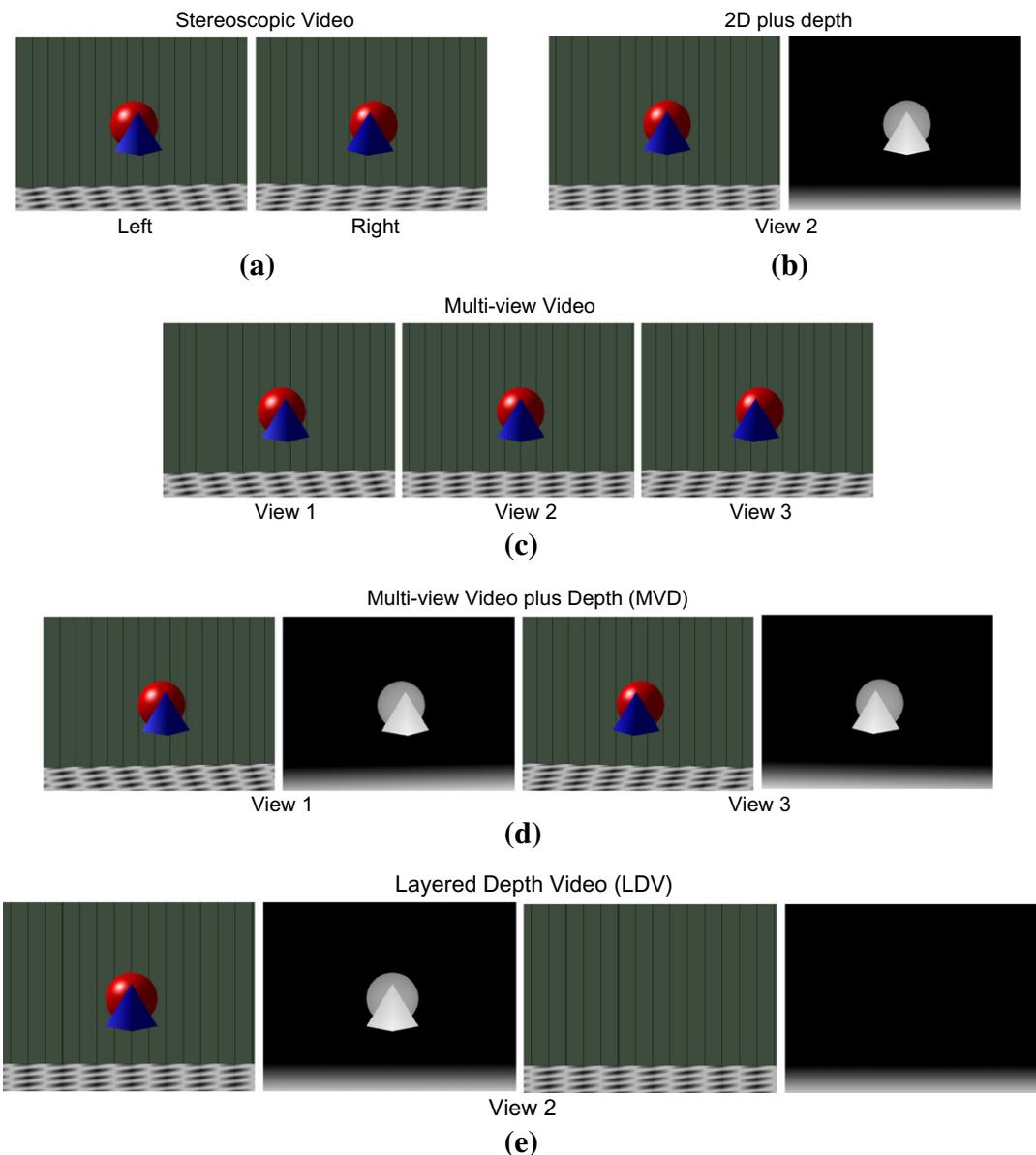
The virtual view is created by setting the value of the pixel at position  $(x_v, y_c)$  in the virtual view equal to the pixel value at position  $(x_c, y_c)$  in the real camera view. This procedure provides satisfactory results for the majority of the pixel positions in the virtual view. However, due to the difference between the two views, some pixel positions will require extra processing to provide a satisfactory virtual view. Some pixels in the camera view will map to the same position in the virtual view. This indicates an occlusion condition so the pixel with the smallest depth value is used in this case. There will also be some positions in the virtual view where no value is mapped from the camera view. These pixel positions are commonly called holes and the process of hole-filling is required to produce a value for these positions. Holes can be caused by a disocclusion condition where parts of the scene which were occluded by foreground objects in the camera view are no longer occluded in the virtual view. Holes will occur along the edge of the view which is visible in the virtual view but not in the camera view and may also be caused by rounding of the values of  $x_v$  to the nearest integer pixel location.

In most situations, the virtual view will be synthesized to provide a view that is between two real camera views. This is known as view interpolation. In this configuration, a virtual view can be synthesized using the real camera views on both sides of the virtual view position. The two synthesized views are then fused to provide the virtual view. View interpolation will provide a pixel value for the majority of the holes in the virtual views produced by each of the camera views separately. Since occluded pixels in one camera view will generally be disoccluded in the adjacent camera view, the holes in the virtual view can be filled using the pixel value from the camera view which provides a non-zero value for that position. In cases where pixels are occluded in both adjacent camera views or where only one camera view is provided, more sophisticated hole-filling techniques are required. These procedures usually involve spatial interpolation of the pixel values from adjacent filled pixel positions.

#### 5.04.5.4 Depth-based 3D video formats

A number of different depth-based representations have been proposed for this approach as shown in [Figure 4.15](#). The simplest of these configurations consists of one camera view and one depth map. The depth map associated with the camera view contains the value of  $z_c$  for each pixel in the camera view. This representation is commonly referred to as 2D + Z. The ISO/IEC 23002-3 specification [[34, 35](#)] (also known as MPEG-C part 3) defines a mechanism for signaling auxiliary depth information associated with a video sequence.

The 2D + Z format can be extended to provide a camera view and a depth map for multiple views. This arrangement is referred to as the Multi-view Video plus Depth (MVD) format. A recently proposed alternative to MVD is known as the Layered Depth Video (LDV) format. This arrangement consists of the camera view and depth map for a single view plus an additional view or views containing depth

**FIGURE 4.15**

D video formats: (a) stereoscopic video, (b) 2D + Z, (c) multi-view video, (d) multi-view video plus depth (MVD), and (e) layered depth video (LDV).

and color information for the background that is occluded in the camera view. The occlusion layer in the LDV format can be generated from the multiple views available in the MVD format. Given left, center, and right camera views, the central view is used to synthesize views from the same view angle as the left and right views. These virtual views will contain holes due to disocclusion that correspond to background pixels in the left and right views. The background pixels from the left and right views that are co-located with the holes in the synthesized views are then warped to their positions in the central view to provide the disocclusion layer information [36].

The MVD and LDV formats can be thought of as extensions to the 2D + Z format with additional views or occlusion information respectively. The extra information provided in these extended formats is used in the view synthesis process when the required information is missing in the 2D + Z format. When the LDV format is used, the view synthesis algorithm generates the required view by first synthesizing the foreground objects at the required position using the depth information for the foreground objects. The disoccluded areas in the background are then filled using the background video and depth information provided in the occlusion layer [37].

In terms of compression efficiency, the two alternative formats have some advantages and disadvantages. The LDV format consists of only one 2D + Z view plus the occlusion information for that view. Alternatively, the MVD format requires at least two views with objects at different positions in the scene according to their associated disparity. To compress the information in this format, the information in one view can be predicted from the other view using disparity compensated inter-view prediction. However, the occlusion information provided in the LDV format cannot be predicted from the single view as, by definition, it does not appear in that view. From a compression perspective, the entropy of the two formats is reduced in different ways. For LDV, the entropy is reduced during the creation of the disocclusion information and further coding will have little effect on the entropy. For the MVD format, correlation exists between the views and this can be exploited to reduce the entropy of the information during the inter-view coding process. In recent experiments, Kerbiriou et al. showed that, for a large baseline framework, LDV significantly outperformed the MVD format [38]. They also showed that using the LDV format could provide satisfactory picture quality for scenes containing transparency and reflection effects.

Since the LDV format contains only one full camera view, this format is incompatible with stereoscopic displays which are not capable of view synthesis. To solve this problem, an alternative format known as Depth Enhanced Stereo (DES) has been proposed [39]. This configuration consists of an LDV arrangement for two views instead of one as is the case with the standard LDV format. The two full views can be used by stereoscopic displays without the need for additional processing. Alternatively, multi-view displays can utilize the extra occlusion information to synthesize views on either side of the stereo camera views. These extrapolated views provide a wider range of views than could be provided with a standard MVD arrangement containing only two views.

#### 5.04.5.5 Generation of content for 3D video

For the 3D video formats described in the previous section, both color and depth information are required. The color information of a scene can be captured using a single video camera or an array of cameras. However, it is usually more difficult to obtain the required depth information for the scene. There are three common approaches that have been adopted to obtain this depth information: depth synthesis from stereo matching, active depth sensing, and 2D–3D conversion from a single view.

#### 5.04.5.5.1 Depth estimation from stereo matching

The estimation of the depth of an object given two views can be explained using the same concepts described in the previous section on view synthesis. Consider the simple case when the two cameras are arranged so that the image planes of the cameras are located in the same plane with only a horizontal shift between them. The relationship between the disparity of the position of a point in the two camera views,  $d_x$  and the depth of this point in the scene,  $z$ , is given by the equation:

$$z = \frac{f_x t_x}{d_x}, \quad (4.15)$$

where  $f_x$  is the horizontal focal length of the camera and  $t_x$  is the horizontal baseline shift between the two cameras. If the cameras are not arranged in this fashion, the camera views can be calibrated to remove the effect of any other translations and rotations and any intrinsic camera effects such as skew and principal point offset. Then, given the two camera views where disparity is inversely proportional to depth, the depth of the objects in the scene can be estimated by determining the disparity of the positions of the object in the two camera views. This process is commonly called stereo matching, since, for each pixel in one view, the matching pixel must be located in the other view. The stereo matching process is similar to that required for inter-view prediction where a window of candidate positions are searched and the pixel that provides the minimum value for some similarity measure is chosen as the best match. The common similarity measures used include sum-of-squared difference (SSD) and normalized cross correlation (NCC). For objects with homogeneous color, noise in the camera acquisition process can cause the searching procedure to produce pixels with similar values but incorrect disparity which leads to a noisy disparity map. To overcome this problem, a smoothness constraint is often added to the similarity measure to form a cost function that is optimized for all pixels in the image. Global optimization strategies that have been used for this type of approach include belief propagation and graph cuts. The resulting disparity map is then converted to a depth map using Eq. (4.15). The quality of the estimated depth map can be improved by additional post-processing to ensure the alignment of depth boundaries with color boundaries and to fill holes caused by occlusion [40].

#### 5.04.5.5.2 Active depth sensing

Even with the adoption of sophisticated global optimization strategies, estimating the depth of objects in a scene using passive stereo matching often fails to produce the correct depth for objects with homogeneous color. These techniques also cannot find the depth for objects that are occluded in either view. One alternative to these passive approaches is to use active depth sensing with physical ranging equipment such as a time-of-flight depth sensor [41] or structured light scanning [42].

The use of active depth sensing however also has associated technical challenges to overcome. The output of depth sensors will typically have a lower spatial resolution than the video camera and hence the depth maps will need to be up-sampled and interpolated to match the color camera view [43]. The video and depth sensors will also be located at different positions and will capture the scene information from different view angles. The resulting depth maps must therefore be synchronized and aligned with its corresponding camera view [44]. Time-of-flight depth sensors also have inherent limitations due to the method used to estimate depth. These sensors use the phase difference between light emitted and the returned reflection of this light from distant objects. Physical aspects of this approach mean that it is limited to a capture range of approximately seven meters and can be affected by noise from ambient

light in outdoor environments. Structured light approaches that use light in the infrared spectrum are similarly affected by interference from ambient infrared light in the scene.

The different limitations of the active and passive approaches are in most cases complementary. Passive approaches have difficulty estimating depth in homogeneous regions but can provide high resolution in textured regions and at the boundaries of objects. Active approaches have lower resolution at depth boundaries but can provide accurate depth in homogeneous regions. Consequently, approaches that fuse the depth maps obtained from both active and passive approaches have also been proposed [45–48].

#### **5.04.5.5.3 2D-3D conversion from a single view**

If only a single camera view is available, depth maps can be estimated using semi-automatic postproduction techniques [49]. This approach can be applied to convert existing content captured in 2D for presentation using 3D display technology. The most common applications for this approach is the conversion of existing 2D movies for display in 3D cinemas. While these techniques usually require some level of manual processing of the content, much of the conversion can be performed automatically using a machine learning approach [50–52]. Training pixels are selected by the user and the algorithm learns the relationship between the relative position, color, and depth of the region in the frame. Since the depth of objects remains relatively constant over time, the algorithms can automatically produce reasonably accurate depth maps for the same object over multiple frames of the single view video. The depths of the objects in the scene are estimated by the manual operator using the monocular depth cues explained previously such as the relative size of objects, perspective and texture gradients, occlusion, shading, specular reflections, and motion parallax. The depth maps estimated using a single camera view can only ever be a coarse planar representation of the true depth of the objects in the scene. However, this amount of depth information is usually enough to provide an adequate perception of the 3D nature of the scene.

#### **5.04.5.5.4 Synthetic views and depth maps from computer generated imagery**

Computer generated imagery (CGI) is another source of 3D video content. Movies created completely using modern CGI animation techniques are rendered into a 2D format via 3D models that describe all the contents in a scene. These 3D model descriptions can easily be converted into 2D camera views and their corresponding depth maps [53]. Combining computer generated content with real video to create a virtual scene is also a popular postproduction technique. Multiple camera views and depth maps can be created in a straightforward way using these techniques. For example, a 3D model of a foreground object can be combined with a real background scene with different disparity to produce views that simulate the different camera views that would be captured if a real foreground object were present. Real video of a segmented foreground object can also be combined with a computer generated background to produce a similar effect.

#### **5.04.5.6 Depth map coding**

Depth map compression is needed to efficiently transmit the depth information required for view synthesis. Depth maps share some inherent information with their corresponding color video frames. However, the characteristics of depth maps differ from the characteristics of color video content. Since depth maps contain only the depth information for an object they do not convey any information about the color and texture of the object. Consequently depth maps are usually modeled as being piecewise smooth. This means that the depth of the surface of an object usually varies smoothly but sharp depth discontinuities

can be present at the boundary between objects and the background. The depth of the surface of an object may not be smooth, e.g., the depth of the leaves on a tree may vary greatly, but in general the variations in the depth of an object are considered to be less than the possible variations in the color of an object.

Video compression algorithms are designed to efficiently code the wide range of textures possible in a video scene. Consequently the compression tools used to compress video luminance and chrominance are not considered to be the most efficient for coding depth maps. Coding algorithms which exploit the piecewise smooth nature of depth maps have been shown to provide a more efficient solution. While depth maps and video content do not have the same characteristics, they do share some common information. The depth discontinuities at boundaries of an object usually coincide with a texture discontinuity in the corresponding video frame. Similarly, since the motion of the objects will produce the same shift in both the depth map and the video frame, the motion compensation required for inter-frame prediction will usually be the same for both types of frames.

There are two other important issues to consider when jointly compressing video with its corresponding depth map. The first point to note is that depth maps are not displayed directly but are used indirectly in view synthesis. This indirect use has implications on how the distortion produced by lossy compression of the depth map will be perceived. Consequently, when optimizing the rate for a given distortion, a joint depth and video coding approach should consider the distortion in the rendered view produced by depth map compression rather than the depth map distortion directly. This approach requires a technique for estimating the rendered view distortion for a given quantization step-size [54]. Estimating the rendered view distortion is a challenging problem since distortion in the depth map will lead to an error in the position of the pixel rather than an error in its color [55]. This type of distortion is commonly referred to as a geometric error. So when estimating the amount of rendered view distortion it is necessary to predict the amount of distortion a geometric error will produce in the synthesized view. This is not a straightforward problem since a geometric error will only cause an error in the synthesized view if the colors of the pixels at the correct and erroneous positions are different. As an example, consider the case of a large region with a completely constant color. In this region, even a large amount of geometric distortion will not produce significant distortion in the synthesized view. Alternatively, geometric errors for pixels located adjacent to a boundary between two objects with large differences in color and depth will almost certainly cause perceptible distortion in the synthesized view [56].

The second issue for consideration is the allocation of the number of bits used for coding the depth map and the video content. This distribution of bits should be carefully balanced so that the perceived distortion caused by lossy compression is approximately equal for the two different types of information. The bit allocation process should again consider the rendered view distortion caused by errors in the depth map and ensure that this distortion is approximately equal to the distortion caused by compressing the luminance and chrominance information of the video content [57].

Platelet coding has been proposed to compress depth maps where the depth map is approximated as piecewise planar images [58,59]. This approach is restricted to planar surfaces so perfect reconstruction is not possible and there is a minimum error associated with this technique. An alternative approach uses transform coding with wavelet basis functions to compress the depth maps.

#### 5.04.5.6.1 Edge-adaptive wavelets

When the high frequency wavelet coefficients are quantized to zero during the compression process the result is ringing artifacts at the boundaries of the objects in the depth map. This information is the most

important to preserve since geometric errors in these regions will result in distortion in the synthesized view. The general solution that has been proposed to solve this problem is to transmit the location of edges as side information to the decoder and then adapt the wavelet transform so that the relatively constant depth information for each object is coded in a different way to the depth information adjacent to edges in the depth map [60,61]. In one proposed approach, the pixels on one side of the edge are symmetrically extended to the other side of the edge to produce a smooth transition and consequently no high frequency information is required. The distortion produced at the boundaries of objects was reduced when compared to a normal wavelet approach using the same number of bits [60]. The edge information can be coded using a differential chain code approach [62]. The reduction in bits for the wavelet coefficients is offset by the bits required to transmit the side information that describes the location of the edges in the depth map. This introduces another bit allocation trade-off for this approach, the amount of bits allocated to the edge information and the wavelet coefficients must be optimized to provide the best synthesized view. An alternative approach is to use wavelet filters with different lengths in different regions of the image. Wavelets with long regions of support can be used for areas far from depth edges for better compression efficiency and wavelets with a shorter region of support, such as the Haar wavelet, in regions adjacent to a depth edge [61].

#### **5.04.5.6.2 Graph-based transforms**

Wavelet coding approaches are not easily implemented into a video coder that uses a block-based architecture such as H.264/AVC. As an alternative to the wavelet approach an edge adaptive block-based transform called the graph-based transform (GBT) was proposed [63–65]. The basic idea in this approach is to transmit a graph-based description of the location of the edges in a block. Each pixel in a block is described as either connected to its immediate neighbors or not, depending on whether there is an edge in between them. This graph-based description is used at both the encoder and decoder to define a set of transform basis functions to code the depth information in the block. Kim et al. showed that this approach provided an approximate 10% savings in bitrate when compared to using the H.264/AVC encoder to compress depth maps.

#### **5.04.5.7 Joint coding of texture and depth**

As the depth and video information share some common properties, strategies to exploit this correlation for increased compression efficiency have been proposed. The first of these techniques is the sharing of motion vector information for prediction of both the depth map and the video content [57]. In this approach the rate distortion optimization of the motion vector choice uses the combined distortion of the video and depth map with a weighting factor to tune the relative importance of the different types of distortion. Another approach involves the use of the edge information in the depth map to provide arbitrarily shaped partitioning information for blocks in the video [66]. Since the edges in the depth map may align with a discontinuity in the motion of objects in the scene, the depth information is used to predict an optimum partitioning of the blocks in the video frame. This arbitrarily shaped partition allows two different sets of motion vectors to accurately predict the different motion of two objects in a block. This approach saves on the bits used for signaling the block partitioning modes and provides a more accurate prediction as the partition is not restricted to lie on the boundary of the sub-partitions in a block. Other approaches include: using the video content to predict the edge locations in the depth

map [67,68] and using the edges in the video frame to improve the accuracy of the edges specified in a compressed description of the depth map edges [69].

## 5.04.6 Quality evaluation of 3D video

Methods for assessing the quality of 2D video are now well established. A standard method for subjective evaluation of television quality video is defined in ITU-R Recommendation BT 500 [70]. The factors that affect the quality of typical 2D video are mainly related to distortion introduced by the lossy compression process. However, for 3D video, there are more factors that influence the perceptual quality of the 3D viewing experience. These extra factors have led to the term quality of experience being used to describe the overall perception of the experience provided by the 3D video [71].

### 5.04.6.1 Sources of perceptual quality degradation in 3D video

For 2D video, highly perceptible distortion caused by a poor compression approach will lead to a low perceptual quality score. For 3D video there are a number of other factors which could lead to a low quality of experience score when viewing the video. These extra factors include inappropriate methods used to create the 3D content, distortion introduced when synthesizing intermediate views for multi-view displays and artifacts introduced by the display technology.

#### 5.04.6.1.1 Perceptual quality degradation introduced during the 3D content creation process

The position of the objects in a scene needs to be carefully considered when creating 3D content [72]. If the objects are positioned too close to the camera, a large amount of negative or crossed disparity will result. This can create a large mismatch between the accommodation and vergence required to view the object in focus which often results in an uncomfortable viewing experience. The objects should also be positioned so that conflicts between monocular and binocular depth cues do not occur. For example consider the case when an object's disparity indicates that its depth is in front of the screen but part of the object lies outside the video frame. It will appear to the viewer that the edge of the display screen is occluding the object which is a monocular depth cue indicating that the object's depth is behind the screen. This conflict is another source of viewer discomfort [73].

Other problems for perceptual quality associated with content creation include the puppet-theater effect and the cardboard effect [74]. While these effects do not usually produce viewer discomfort they do result in objects in the scene that have an unnatural appearance. The puppet-theater effect occurs when a toed-in camera configuration is used and causes objects in the foreground of the scene to appear unnaturally small. The cameras are said to be toed-in if the lines normal to the image plane in the two pinhole camera models are not parallel but instead meet at a point some distance in front of the cameras. This problem can be solved by using a camera configuration where the lines normal to the image planes in the two cameras are parallel. The cardboard effect occurs when the field of view of the cameras is significantly smaller than the field of view provided to the viewer by the display. This mismatch causes a lateral shift in the position of an object to be magnified compared to a change in the depth of the object and produces objects that appear to have only one depth. Both foreground and background objects appear to be positioned at the correct depths but look unnaturally flat.

#### 5.04.6.1.2 View synthesis distortion

As mentioned previously, distortion introduced by the data compression process applied to the **depth map** will result in geometric errors that will impact the quality of the view synthesis process. These geometric errors will typically produce noticeable **warping of the shape** of objects. Other types of distortion introduced in the view synthesis process include: **blurring** of disoccluded background regions during interpolation from adjacent regions, failure of the **hole-filling** algorithms to synthesize complex background textures and **flickering** along the boundaries of objects caused by temporal inconsistencies in the location of synthesized objects.

#### 5.04.6.1.3 Distortion caused by the display

The main source of perceptual quality degradation caused by the display device is crosstalk between the views for the left and right eyes. This occurs when the glasses used in stereoscopic displays do not completely filter out the view meant for the opposite eye or when the viewer's eyes are positioned incorrectly when viewing multi-view autostereoscopic displays.

### 5.04.6.2 Standards and metrics for quality evaluation of 3D video

The standard for assessing the subjective quality of 3D video, ITU-R Recommendation BT 1438 [75], closely follows the ITU-R Recommendation BT 500 standard for assessing 2D video [76]. Consequently, this standard is generally considered to be **inadequate** for assessing the full 3D quality of experience [77, 71]. Hence, an extended version of this standard is required that includes methods for assessing the quality of depth perception and visual comfort.

Objective metrics have also been developed for estimating picture quality and providing a score similar to the subjective rating [78]. For 3D video these scores are usually evaluated by comparing the values produced with the scores from subjective tests which use the methods defined in ITU-R Recommendation BT 500. Consequently these objective measures also do not capture the quality of the overall 3D viewing experience including the **quality of the depth perception** and **viewing comfort**. Researchers are beginning to address this problem, for example, in [79] mathematical models were derived to explain the just noticeable difference in depth (JNDD) using binocular disparity, retinal blur, and relative size as depth cues.

### 5.04.7 Conclusions

Display technologies and methods for content creation for 3D video have improved markedly in recent times. These improvements have led to a rapid increase in the popularity of 3D entertainment services such 3D cinema and 3D television.

Currently, stereoscopic video is the **format** used in 3D Cinema and 3D TV. However, future display technologies will provide a more realistic viewing experience by providing multiple views. The increased number of views required for these new display technologies mean that data compression will be an essential part of future broadcast and storage applications. Since the number of views provided by the different display technologies is variable, the **decoupling** of the data format and the display technology is seen as an important **requirement** for any future multi-view system. **Data formats that include depth**

have been proposed as a solution that will allow this decoupling. Current compression techniques for these depth-based formats are in the early stages of development. Areas that require further research in this domain include: the **combined compression of the depth, motion and video components of depth-based formats**, **improved view synthesis algorithms**, and **encoder optimization approaches** that use the distortion of the synthesized views rather than the depth map distortion directly.

Improved standards for assessing the subjective quality of 3D video and objective metrics for measuring the quality of the 3D video viewing experience are also required. The accurate assessment of the quality of the 3D viewing experience is essential for providing feedback to optimize the encoding algorithms for 3D video.

## Relevant Websites

3D4YOU—Content Generation and Delivery for 3D Television. <http://www.hitech-projects.com/euprojects/3d4you/www.3d4you.eu/index-2.html>.

3D Innovation Center. <http://www.3dinnovationcenter.de/en/>.

Disney Research—Video Processing. <http://www.disneyresearch.com/research-areas/video-processing/>.

The World of 3D Imaging. <http://www.stereoscopy.com/>.

The History of Stereo Photography. [http://www.arts.rpi.edu/~ruiz/stereo\\_history/text/historystereog.html](http://www.arts.rpi.edu/~ruiz/stereo_history/text/historystereog.html).

## Glossary

### **proprioception**

how the brain senses the spatial orientation and movement of parts of the body using only stimuli from brain signals that control muscle movement

### **vergence**

the amount of rotation the eyes undergo to place the light rays emitted from an object at the same point on the fovea of both left and right eyes

### **accommodation**

the change in focal length required by the lens in the front of the eye to keep the object of interest in focus

### **horopter**

the theoretical locus of points, for a given vergence angle, where objects located at these points will be perceived as a single object

### **Panum's fusional area**

a region surrounding the horopter in which objects are perceived as a single object

### **diplopia**

when an object is located outside Panum's fusional area, light rays from the object will fall on different parts of the retina in each eye and the object is perceived as two objects

### **parallax**

the relative position of an object in different views of the same scene

### **motion parallax**

the change in the relative position of an object when the viewer's position moves relative to the scene

### **stereoscopic display**

a display where the view for both the left and right eye is displayed simultaneously on the screen or monitor and the viewer is required to wear special glasses which allow only the appropriate view to be seen by each eye

<b>autostereoscopic display</b>	a display which provides a mechanism for the left and right views to be seen by the corresponding eye of the viewer without the need for special glasses
<b>multi-view display</b>	an autostereoscopic display where multiple adjacent views of the same 3D scene are provided in the viewing zone. As the viewer's eyes move to the left and right, the view changes to provide a different perspective of the same scene
<b>super multi-view display</b>	a display where the number of views is sufficient to produce the effect of continuous motion parallax
<b>quality of experience</b>	the overall perception of the experience provided by 3D video

---

## References

- [1] W. Ijsselsteijn, H. de Ridder, R. Hamberg, D. Bouwhuis, J. Freeman, Perceived depth and the feeling of presence in 3DTV, *Displays* 18 (1998) 207–214.
- [2] B. Mendiburu, *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*, Focal Press/Elsevier, 2009.
- [3] I.P. Howard, B.J. Rogers, *Perceiving in depth*, *Stereoscopic Vision*, vol. 2, Oxford University Press, USA, 2012.
- [4] P.A. Howarth, Potential hazards of viewing 3-D stereoscopic television, cinema and computer games: a review, *Ophthal. Physiol. Opt.* 31 (2011) 111–122.
- [5] M.T.M. Lambooij, W.A. Ijsselsteijn, I. Heynderickx, Visual discomfort in stereoscopic displays: a review, in: *Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems XIV*, 64900I, San Jose, CA, 6490, 2007.
- [6] C. Wheatstone, Contributions to the physiology of vision. Part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision, *Philos. Trans. R. Soc. Lond.* 128 (1838) 371–394.
- [7] P. Benzie, J. Watson, P. Surman, I. Rakkolainen, K. Hopf, H. Urey, V. Sainov, C. von Koplow, A survey of 3DTV displays: techniques and technologies, *IEEE Trans. Circ. Syst. Video Technol.* 17 (2007) 1647–1658.
- [8] B.G. Blundell, A.J. Schwarz, *Volumetric Three-Dimensional Display Systems*, Wiley-IEEE Press, New York, 2000.
- [9] H. Urey, K.V. Chellappan, E. Erden, P. Surman, State of the art in stereoscopic and autostereoscopic displays, *Proc. IEEE* 99 (2011) 540–555.
- [10] W. Rollmann, Zwei neue stereoskopische methoden, *Annalen der Physik* 166 (1853) 186–187.
- [11] I. Ideses, L. Yaroslavsky, Three methods that improve the visual quality of colour anaglyphs, *J. Opt. A Pure Appl. Opt.* 7 (2005) 755.
- [12] H. Jorke, A. Simon, M. Fritz, Advanced stereo projection using interference filters, in: *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, IEEE, 2008, pp. 177–180.
- [13] I. Sexton, P. Surman, Stereoscopic and autostereoscopic display systems, *Signal Process. Mag. IEEE* 16 (1999) 85–99.
- [14] J. Konrad, M. Halle, 3-D displays and signal processing, *Signal Process. Mag. IEEE* 24 (2007) 97–111.
- [15] Y. Takaki, High-density directional display for generating natural three-dimensional images, *Proc. IEEE* 94 (2006) 654–663.
- [16] C. Moller, A. Travis, Flat panel time multiplexed autostereoscopic display using an optical wedge waveguide, in: *Proceedings of the 11th International Display Workshops (IDW 04)*, Society for Information Display, 2004.
- [17] M. Cho, M. Daneshpanah, M. Inkyu, B. Javidi, Three-dimensional optical sensing and visualization using integral imaging, *Proc. IEEE* 99 (2011) 556–575.

- [18] M. Miura, J. Arai, T. Mishina, M. Okui, F. Okano, Integral imaging system with enlarged horizontal viewing angle, in: Proc. of SPIE 8384, Three-Dimensional Imaging, Visualization, and Display, 2012, p. 83840O, (May 1, 2012), doi: <http://dx.doi.org/10.1117/12.921388>.
- [19] W.-L. Chen, C.-H. Tsai, C.-S. Wu, C.-Y. Chen, S.-C. Cheng, A high-resolution autostereoscopic display system with a wide viewing angle using an LCOS projector array, *J. Soc. Inf. Dis.* 18 (2010) 647–653.
- [20] D. Brewster, *The Stereoscope; its History, Theory, and Construction*, Hastings on Hudson, New York, 1856.
- [21] R.M. Hayes, *3D Movies: A History and Filmography of Stereoscopic Cinema*, McFarland, New York, 1998.
- [22] P. Angueira, D. de la Vega, J. Morgade, M.M. Velez (Eds.), *Transmission of 3D Video Over Broadcasting*, Springer, New York, 2012.
- [23] C. Fehn, 3D TV broadcasting, in: O. Schreer, P. Kauff, T. Sikora (Eds.), *3D Videocommunication*, Wiley, UK, 2005.
- [24] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, C. Zhang, Multiview imaging and 3DTV, *IEEE Signal Process. Mag.* 24 (2007) 10–21.
- [25] M. Tanimoto, Overview of free viewpoint television, *Signal Process. Image Commun.* 21 (2006) 454–461.
- [26] M. Tanimoto, M.P. Tehrani, T. Fujii, T. Yendo, Free-viewpoint TV, *Signal Process. Mag. IEEE* 28 (2011) 67–76.
- [27] A. Smolic, 3D video and free viewpoint video—from capture to display, *Pattern Recognit.* 44 (2011) 1958–1968.
- [28] G.J. Sullivan, J. Ohm, H. Woo-Jin, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Trans. Circ. Syst. Video Technol.* 22 (2012) 1649–1668.
- [29] Information Technology—Coding of Audio-Visual objects, Part 10: Advanced Video Coding, ISO/IEC 14496-10:2012(E), 2012.
- [30] A. Vetro, T. Wiegand, G.J. Sullivan, Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard, *Proc. IEEE* 99 (2011) 626–642.
- [31] P. Merkle, A. Smolic, K. Müller, T. Wiegand, Efficient prediction structures for multiview video coding, *IEEE Trans. Circ. Syst. Video Technol.* 17 (2007) 1461–1473.
- [32] Y.K. Park, K. Jung, Y. Oh, S. Lee, J.K. Kim, G. Lee, H. Lee, K. Yun, N. Hur, J. Kim, Depth-image-based rendering for 3DTV service over T-DMB, *Signal Process. Image Commun.* 24 (2009) 122–136.
- [33] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability, *Signal Process. Image Commun.* 22 (2007) 217–234.
- [34] A. Bourge, J. Gobert, F. Bruls, MPEG-C Part 3: enabling the introduction of video plus depth contents, in: Proc. of IEEE Workshop on Content Generation and Coding for 3D-television, Eindhoven, The Netherlands, 2006.
- [35] Information Technology—MPEG Video Technologies, Part 3: Representation of Auxiliary Video and Supplemental Information ISO/IEC 23002-3:2007(E), 2007
- [36] B. Bartczak, P. Vandewalle, O. Grau, G. Briand, J. Fournier, P. Kerbiriou, M. Murdoch, M. Müller, R. Goris, R. Koch, Display-independent 3D-TV production and delivery using the layered depth video format, *IEEE Trans. Broadcasting* 57 (2011) 477–490.
- [37] K. Müller, A. Smolic, K. Dix, P. Kauff, T. Wiegand, Reliability-based generation and view synthesis in layered depth video, in: IEEE 10th Workshop on Multimedia Signal Processing, 2008, IEEE, 2008, 34–39.
- [38] P. Kerbiriou, G. Boisson, K. Sidibé, Q. Huynh-thu, Depth-based representations: which coding format for 3D video broadcast applications? in: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2011, 78630D–78630D-10.
- [39] A. Smolic, K. Müller, P. Merkle, P. Kauff, T. Wiegand, An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution, in: Picture Coding Symposium, PCS 2009, IEEE, 2009, pp. 1–4.

- [40] A. Smolic, P. Kauff, S. Knorr, A. Hornung, M. Kunter, M. Muller, M. Lang, Three-dimensional video post-production and processing, Proc. IEEE 99 (2011) 607–625.
- [41] R. Lange, P. Seitz, Solid-state time-of-flight range camera, IEEE J. Quant. Electron. 37 (2001) 390–397.
- [42] D. Scharstein, R. Szeliski, High-accuracy stereo depth maps using structured light, in: IEEE Computer Society Conference on Proceedings of the Computer Vision and Pattern Recognition, 2003, vol. 1, IEEE, 2003, pp. I-195–I-202.
- [43] Q. Yang, R. Yang, J. Davis, D. Nistér, Spatial-depth super resolution for range images, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07, IEEE, 2007, pp. 1–8.
- [44] J. Zhu, L. Wang, R. Yang, J. Davis, Fusion of time-of-flight depth and stereo for high accuracy depth maps, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, IEEE, 2008, pp. 1–8.
- [45] Q. Yang, K.H. Tan, B. Culbertson, J. Apostolopoulos, Fusion of active and passive sensors for fast 3D capture, in: IEEE International Workshop on Multimedia Signal Processing, 2010.
- [46] B. Bartczak, R. Koch, Dense depth maps from low resolution time-of-flight depth and high resolution color views, Adv. Vis. Comput. (2009) 228–239.
- [47] Q. Li, M. Biswas, M.R. Frater, M.R. Pickering, Phase based disparity estimation using adaptive structured light and dual-tree complex wavelet, in: International Conference on Digital Image Computing Techniques and Applications (DICTA), 2011, IEEE, 2011, pp. 78–83.
- [48] Q. Li, M. Biswas, M.R. Pickering, M.R. Frater, Accurate depth estimation using structured light and passive stereo disparity estimation, in: 18th IEEE International Conference on Image Processing (ICIP), 2011, IEEE, 2011, pp. 969–972.
- [49] W.J. Tam, L. Zhang, 3D-TV content generation: 2D to 3D conversion, in: IEEE International Conference on Multimedia and Expo, 2006, IEEE, 2006, pp. 1869–1872.
- [50] P. Harman, J. Flack, S. Fox, M. Dowley, Rapid 2D to 3D conversion, in: Proc. SPIE, 2002.
- [51] W.J. Tam, F. Speranza, S. Yano, K. Shimono, H. Ono, Stereoscopic 3D-TV: visual comfort, IEEE Trans. Broadcasting 57 (2011) 335–346.
- [52] L. Zhang, C. Vázquez, S. Knorr, 3D-TV content creation: automatic 2D to 3D video conversion, IEEE Trans. Broadcasting 57 (2011) 372–383.
- [53] A.A. Alatan, Y. Yemez, U. Gudukbay, X. Zabulis, K. Muller, C.E. Erdem, C. Weigel, A. Smolic, Scene representation technologies for 3DTV—a survey, IEEE Trans. Circ. Syst. Video Technol. 17 (2007) 1587–1605.
- [54] W.S. Kim, A. Ortega, P.L. Lai, D. Tian, C. Gomila, Depth map coding with distortion estimation of rendered view, SPIE Vis. Inf. Process. Commun. 7543 (2010) 75430B.
- [55] G. Tech, K. Muller, T. Wiegand, Evaluation of view synthesis algorithms for mobile 3DTV, in: 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON), 2011, pp. 1–4, 16–18 May.
- [56] W.S. Kim, A. Ortega, J. Lee, H.C. Wey, 3-D video coding using depth transition data, in: Picture Coding Symposium (PCS), 2010, IEEE, 2010, pp. 178–181.
- [57] I. Daribo, C. Tillier, B. Pesquet-Popescu, Motion vector sharing and bitrate allocation for 3D video-plus-depth coding, EURASIP J. Appl. Signal Process. 2009 (2009) 3.
- [58] R.M. Willett, R.D. Nowak, Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging, IEEE Trans. Med. Imaging 22 (2003) 332–350.
- [59] Y. Morvan, D. Farin, Platelet-based coding of depth maps for the transmission of multiview images, in: Electronic Imaging 2006, International Society for Optics and Photonics 2006, pp. 60550K–60550K-12.
- [60] M. Maitre, Y. Shinagawa, M.N. Do, Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering, IEEE Trans. Image Process. 17 (2008) 946–957.
- [61] I. Daribo, C. Tillier, B. Pesquet-Popescu, Adaptive wavelet coding of the depth map for stereoscopic view synthesis, in: IEEE 10th Workshop on Multimedia Signal Processing, IEEE, 2008, pp. 413–417.

- [62] H. Freeman, On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Comput.* (1961) 260–268.
- [63] G. Shen, W.S. Kim, S. Narang, A. Ortega, J. Lee, H. Wey, Edge-adaptive transforms for efficient depth map coding, in: Picture Coding Symposium (PCS), 2010, IEEE, 2010, pp. 566–569.
- [64] G. Shen, W.S. Kim, A. Ortega, J. Lee, H. Wey, Edge-aware intra prediction for depth-map coding, in: ICIP, 2010, pp. 3393–3396.
- [65] W.S. Kim, S.K. Narang, A. Ortega, Graph based transforms for depth video coding, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, IEEE, 2012, pp. 813–816.
- [66] I. Daribo, D. Florencio, G. Cheung, Arbitrarily shaped sub-block motion prediction in texture map compression using depth information, in: Picture Coding Symposium (PCS), 2012, IEEE, 2012, pp. 121–124.
- [67] P. Merkle, C. Bartnik, K. Muller, D. Marpe, T. Wiegand, 3D video: depth coding based on inter-component prediction of block partitions, in: Picture Coding Symposium (PCS), 2012, IEEE, 2012, pp. 149–152.
- [68] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Muller, H. Rhee, 3D video coding using advanced prediction, depth modeling, and encoder control methods, in: Picture Coding Symposium (PCS), 2012, IEEE, 2012, pp. 1–4.
- [69] L. Yu, M. Tanimoto, C. Zhu, Y. Zhao, *3D-TV System with Depth-Image-Based Rendering: Architecture, Techniques and Challenges*, Springer, 2012.
- [70] ITU-R Recommendation BT.500-13, Methodology for the Subjective Assessment of the Quality of Television Pictures, 2012.
- [71] M. Barkowsky, K. Brunnstrom, T. Ebrahimi, L. Karam, P. Le Callet, A. Perkis, A. Raake, M. Subedar, K. Wang, L. Xing, J. You (Eds.), *Subjective and Objective Visual Quality Assessment in the Context of Stereoscopic 3D-TV*, Springer, New York, 2012.
- [72] F. Zilly, J. Kluger, P. Kauff, Production rules for stereo acquisition, *Proc. IEEE* 99 (2011) 590–606.
- [73] T. Shibata, J. Kim, D.M. Hoffman, M.S. Banks, The zone of comfort: predicting visual discomfort with stereo displays, *J. Vis.* 11 (2011).
- [74] H. Yamanoue, M. Okui, F. Okano, Geometrical analysis of puppet-theater and cardboard effects in stereoscopic HDTV images, *IEEE Trans. Circ. Syst. Video Technol.* 16 (2006) 744–752.
- [75] ITU-R Recommendation BT.1438, Subjective assessment of stereoscopic television pictures, 2000.
- [76] K. Wang, M. Barkowsky, K. Brunnstrom, M. Sjostrom, R. Cousseau, P. le Callet, Perceived 3D TV transmission quality assessment: multi-laboratory results using absolute category rating on quality of experience scale, *IEEE Trans. Broadcasting* (2012) 1.
- [77] L.M.J. Meesters, W.A. Ijsselsteijn, P.J.H. Seuntiens, A survey of perceptual evaluations and requirements of three-dimensional TV, *IEEE Trans. Circ. Syst. Video Technol.* 14 (2004) 381–391.
- [78] A. Benoit, P. le Callet, P. Campisi, R. Cousseau, Quality assessment of stereoscopic images, *EURASIP J. Image Video Process.* 2008 (2008) 659024.
- [79] V. de Silva, A. Fernando, S. Worrall, H.K. Arachchi, A. Kondoz, Sensitivity analysis of the human visual system for depth cues in stereoscopic 3-D displays, *IEEE Trans. Multimedia* 13 (2011) 498–506.

# Perceptually Optimized Video Compression

# 5

**Matteo Naccari and Marta Mrak**

*British Broadcasting Corporation – Research and Development, 56 Wood Lane, W12 7SB London, UK*

---

## Nomenclature

$f$	denotes the continuous frequency in the spatial or temporal domain
$C$	denotes the frequency coefficient relative to a specific transformation (e.g., the Discrete Cosine Transform)
$(i,j)$	denotes the position of a <b>frequency coefficient</b> in the $i$ th row and $j$ th column inside a block of transformed coefficients
$(x,y)$	denotes the position of an image pixel located at row $x$ and column $y$
$JND(\cdot)$	denotes the Just Noticeable Distortion value associated to a given coordinate expressed by the argument $(\cdot)$ . The coordinate can be the frequency coefficient or the pixel position
$l$	denotes the <b>luminance</b> value for one or a set of image pixels
$E(\cdot)$	denotes the <b>energy</b> for a set of frequency coefficients and measured as the sum of their square values
$B$	denotes the number of samples in a row/column of one image block of coefficients or pixels
$\mu$	denotes the average pixel <b>intensity</b> (luminance or chrominance)
$dB$	the decibel is a logarithmic unit that is used to measure a ratio between two signals, for example, the peak signal-to-noise ratio (PSNR) such as the noise introduced by compression of the original (reference) signal
$bps$	the bit-per-second is a unit that quantifies bitrate, thus representing the average number of bits required for transmission of a video sequence per unit of time. Usually, the bps is also reported in multiples of the kilo (k). In this chapter kilo bps (kbps) is used

---

### 5.05.1 Introduction

In recent years, High Definition Television (HDTV) services have become more popular in broadcasting and other applications, while the development of technologies supporting Ultra High-Definition Television (UHDTV) [1] has significantly progressed. These formats with higher spatial and temporal resolutions have been introduced mainly to provide viewers with a more immersive quality of experience. An example of a system that employs the UHD format is the Super Hi-Vision (SHV), under development by the Japanese broadcaster NHK which will target  $7680 \times 4320$  spatial resolution at 120

frames per second temporal resolution [2]. The SHV format is still in its experimental and development stage although it has already been used to shoot events related to the 2012 Summer Olympics in London [3]. Given the emerging use of this UHD format in practice, it is expected that the need to reduce the storage space and transmission bandwidth will become more significant in the coming years.

The solution to the problem of bandwidth and storage space reduction is provided by video compression which can be utilized in a way that minimizes the coded bitrate for a given target quality or maximizes the video quality under a rate constraint. In the past 30 years, research in video coding has provided successful schemes which have been eventually standardized as, for example in cases such as MPEG-2/H.262 [4] and H.264/AVC [5]. Improved and efficient video coding tools have effectively reduced the coding bitrate by a factor of two (at comparable visual quality) every 10 years when a new standard becomes available [6]. This trend is expected to be followed by the new High Efficiency Video Coding (HEVC) standard [7], jointly being standardized by the ISO/IEC Moving Picture Expert Group (MPEG) and the ITU-T Video Coding Expert Group (VCEG).

All video codecs belonging to the MPEG-x and/or H.26x families are block-based hybrid motion compensated predictive coders which achieve high compression by employing lossy quantization in the transform domain. The overall effect of lossy compression in block-based video coders is the introduction of artifacts in the decoded video. These are normally noticeable as false edges along the block boundaries where the quantization is applied (i.e., blockiness) or as a loss of picture detail (i.e., blurring) [8]. To demonstrate the usual appearance of coding artifacts, an example is provided in Figure 5.1. The first frame of a popular test video sequence *Foreman* in CIF resolution has been coarsely compressed with a fixed quantization step. Compared to the original frame (left), coding artifacts in the compressed frame consist of blocking and strong blurring. In this case the blurring causes the loss of texture detail on the building wall in the background. During compression the quantization step was uniform across the whole frame, i.e., each image area was equally quantized without considering the underlying content. However, the perceived quality varies significantly across the frame. While the coding artifacts around the face of the actor are most disturbing, the errors introduced in the building wall in the background are less visible. Figure 5.1 is not only representative of common artifacts introduced by lossy coding, but also demonstrates properties of the Human Visual System (HVS) which, as observed above, has a varying spatial (and temporal) sensitivity to these artifacts. Since viewers are the ultimate

观察结果  
表现形式



**FIGURE 5.1**

First frame of the *Foreman* CIF sequence. (Left) Original frame. (Right) The same frame coarsely quantized with a constant quantization step.

“*quality judges*” of coded videos, it is beneficial to consider different properties of the HVS during the whole video coding process.

This need was initially observed in the early 1990s [9] and reinforced in subsequent studies [8]. Unfortunately however, state-of-the-art video codecs still do not fully exploit the HVS properties when encoding a video. A typical example to demonstrate this practice is related to the coding mode selection for image blocks within frames. Depending on the frame type, several coding modes are available and, to select the best one, video encoders perform a **Rate-Distortion Optimization (RDO)** search by minimizing a given cost function [10]. Such a cost function is a combination of **coding rate** and **distortion**, where the latter is usually measured as the **Mean Square Error** (MSE) between the pixel values in the original image block and the pixels in the reconstructed block after decoding. MSE has been widely used in image and video coding because it has a **simple** formulation, it is **mathematically tractable** (namely for **solving convex optimization problems**), and it is **additive**. However, it is well known that MSE **poorly correlates** with the **perceived quality** [11,12]. To demonstrate some of the shortcomings of MSE, **Figure 5.2** shows the original *Lena* image on the left and its one-pixel shifted version on the right. The two images look identical. However, considering 8 bits per component, the MSE measured on the luma component of the whole image has a value of 124. By definition, the MSE measures the square distance between the original and the processed image, therefore the closer to zero the MSE score is, the more similar the processed image is to the original. In this example, a value (124) far away from zero would lead to score the image on the right as of much worse quality with respect to the original. This example shows that imperceptible processing of a given image may lead to very high MSE values which can negatively influence the performance of compression algorithms that rely on this distortion metric. In another example, **Figure 5.3**, the image on the left has been tampered by inserting a patch of gray pixels. From that image the MSE is measured with respect to the original image. The MSE obtained (11 in this case) is used as the variance of a white Gaussian noise added to the original *Lena* image to obtain the picture on the right in **Figure 5.3**. This time, the two images have the same MSE but the perceived

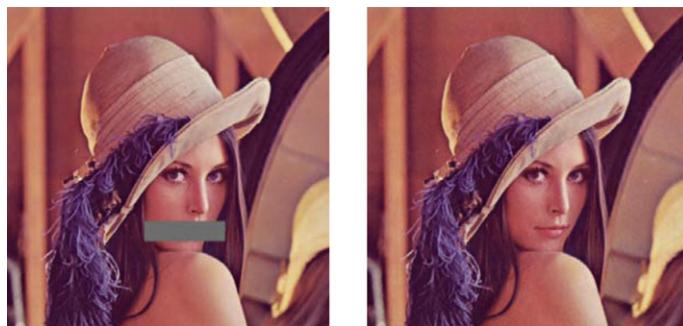


**FIGURE 5.2**

Example of MSE shortcoming. (Left) Original *Lena* image. (Right) *Lena* image shifted on the left by one pixel. Though the two images look identical, the measured MSE on the luma component suggests a large difference.

使用原因

与实际相关性

**FIGURE 5.3**

Example of another MSE shortcoming. (Left) *Lena* image distorted by applying a plain patch of gray pixels which led to a MSE value of 11 in the luma component when compared to the original. (Right) *Lena* image distorted by adding a space invariant Gaussian noise whose variance is still equal to 11.

不足

quality is significantly different. This example demonstrates that the MSE can be blind with respect to the type of distortion added to the image and video content.

优点

To overcome the aforementioned MSE shortcomings, several alternative quality metrics have been developed for image and video coding applications (the interested reader is referred to [13, 14], and references therein for a thoughtful state-of-the-art review on this topic). Although these metrics provide a good prediction of the perceived quality, they are either very complex to compute or not additive which makes their integration in practical video coding schemes unfeasible. A popular example among such alternative image quality metrics is the Structural SIMilarity (SSIM) metric [15]. This metric represents a good trade-off between perceived quality prediction and computational complexity and is also additive. For these reasons it has been integrated in video coding architectures [14, 16] to improve the video quality under given rate constraints. However, the SSIM metric is not convex which complicates finding closed form solutions to minimization problems. Moreover, the sensitivity of SSIM with respect to quality variations is still not well understood. In fact, provided that the SSIM metric scores the image quality in the range  $[0, 1]$  where 1 means identical quality with respect to the reference, it is still not clear what is the score change which allows a noticeable quality difference when the test image is compared to the reference.

劣

The previous discussion and examples make clear the need for video coding schemes more “perceptually tuned” and which explicitly take into account the HVS properties. The advantage brought by the integration of the HVS properties in a video codec is twofold: first the quality of coded videos is perceptually maximized and secondly, by taking advantage of the space and time varying distortion sensitivity of the HVS, the coding rate can be reduced for a given target quality. The area which studies and devises solutions to integrate HVS properties in video codecs is often referred to as *perceptual video coding* [8, 9].

In this context, this chapter provides an overview of the models proposed in the literature for the HVS sensitivity to coding artifacts and how they have been or can be integrated in video coding architectures. In the modeling of the HVS sensitivity to distortion, relevant approaches from the literature

are presented and reviewed. The discussion on **model integration** has a **threefold goal**: first describing the functionalities offered by video coding standards to integrate perceptual mechanisms in the coding process, second reviewing the existing perceptual video coding solutions, and finally discussing how video codecs can be modified to accommodate HVS models also taking into account complexity constraints. Moreover, beside perceptual video codecs which explicitly integrate human visual system models, the review also summarizes other video coding schemes which still take advantage of the HVS characteristics but follow a different approach whereby some image areas are not encoded and then synthesized at the decoder.

基础

Throughout this chapter, the block-based motion compensated hybrid codec architectures standardized by the MPEG-x and H.26x families are assumed as the **reference video coding scheme**. The reason for limiting the review to these architectures is due to their widespread popularity in current video coding applications. For such codecs, a **prediction** (either spatial or temporal) is formed for each image block being coded and it is subtracted from the original data. The difference obtained (also called the **residual**) is then **transformed**, **quantized**, and **entropy** coded. The transformation and quantization processes are applied over a non-overlapping grid of  $B \times B$  blocks where  $B$  varies according to the standard specification.<sup>1</sup> Moreover, the transformation, i.e., the tool for spatial frequency decomposition, is performed using approximations of the 2D Discrete Cosine Transform (DCT). Therefore, the following discussion will focus only on video codecs which apply quantization in the DCT domain.

The remainder of this chapter is thus structured as follows. **Section 5.05.2** describes the **main coding stages** which can be modified to perform perceptual video coding and provides some **general considerations** for the design of perceptual coding tools. Moreover, it introduces and defines the **Just Noticeable Distortion (JND) paradigm** which is then used to model the HVS sensitivity to coding artifacts. In **Section 5.05.3** the HVS **masking phenomena** which contribute to the overall JND for a given video content are discussed and the state-of-the-art models proposed in the literature are presented. **Section 5.05.4** discusses the main issues related to the integration of JND models in video codecs and presents the normative functionalities offered by some video coding standards to allow the integration of JND models in video coding architectures. Conversely, in **Section 5.05.5** several perceptual video coding schemes proposed in the literature are reviewed and discussed with particular emphasis on the computational complexity involved during the integration of human visual system models in the coding process. Finally, **Section 5.05.6** draws the conclusions and points out some future research directions.

因果  
描述

## 5.05.2 Perceptual coding tools in video coding architectures

Through examples given in the Introduction, it has been shown that the human visual system perceives coding artifacts differently dependent on the **characteristics** of the video sequence. From this observation one can devise a **perceptual video coding scheme** which applies coarser compression to image areas where the human visual system is less sensitive to artifacts and a finer compression otherwise. In order to accomplish this goal, there are **two main questions** which need to be addressed: first **which coding modules** can be considered for perceptual video coding and, second, **how to design** perceptual video

<sup>1</sup>For the MPEG-2 standard  $B = 8$  and for the H.264/AVC standard  $B$  may be equal to 4 or 8. The emerging HEVC standard also allows use of 16 and 32 as block sizes.

coding tools. This section answers these two questions by making general observations about the video coding process and presenting two common design approaches for perceptual video coding tools.

### 5.05.2.1 Perceptual optimization of coding modules

In block-based motion compensated video coding architectures, three coding stages are commonly tailored to optimize coded video quality from a perceptual point of view. These are: quantization, in-loop filter, and rate-distortion optimization.

**组成** Quantization consists of scaling or discarding transform coefficients related to the prediction residuals. More precisely, each prediction residual is scaled down as defined by a quantization step to provide an approximation that is more efficiently compressed. To obtain the reconstructed values, inverse scaling is used. By varying the quantization step size, the bitrate needed to represent video content is reduced while coding artifacts are inevitably introduced as the aforementioned scaling is irreversible. It should be noted that quantization is the main source of distortion introduced in the video compression process. Therefore it plays a key role, not only in video coding but also in the perceptual optimization of video codecs.

In-loop filters aim to improve the perceived video quality by reducing coding artifacts. Generally these filters also improve objective quality by reducing the MSE between the original and the reconstructed video. A well-known example of in-loop filter is the deblocking filter [17] used in both H.264/AVC and HEVC standards [18]. This filter operates along the coding block boundaries and applies a different amount of filtering depending on the local image features and coding parameters. More precisely, the filter acts to reduce the blocking artifacts while preserving image edges. Given its quality enhancement nature, an in-loop filter can be optimized with respect to perceptual metrics to provide a better perceived quality.

As already mentioned in the Introduction, RDO selects the best coding mode for each image block by minimizing a rate-distortion cost function. Generally, the distortion function is assumed to be Mean Square Error (MSE) but, by changing the MSE with a quality metric well correlated with the perceived quality, the overall coding process can be tailored toward a better subjective quality. Alternatively, the coding rate can be minimized by selecting coding modes which provide the same perceived quality but with fewer bits. Thus the RDO module offers scope to integrate perceptual models and can be used to design perceptually tuned video coding solutions.

In Section 5.05.4.4 these three coding modules will be discussed again with the emphasis on the normative functionalities provided by state-of-the-art video coding standards to enable the integration of perceptual models and tools in the coding process.

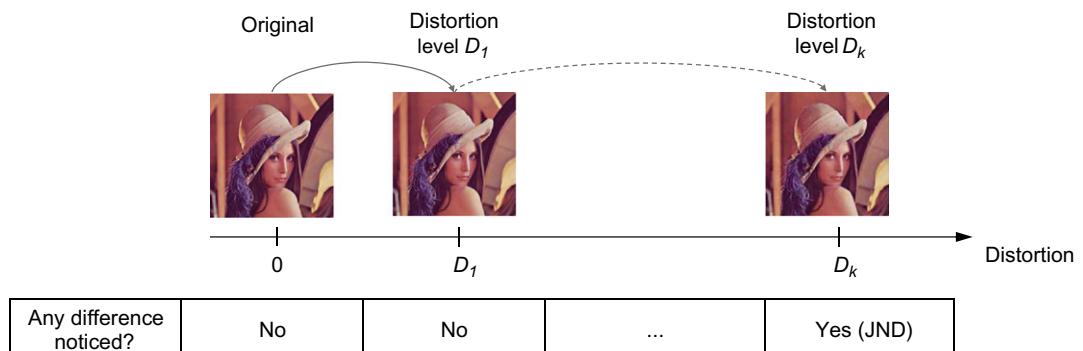
### 5.05.2.2 Design of perceptual video coding tools

As stated in the Introduction, the main goal of perceptual video coding is to design tools that can be integrated in video codecs leading to improved subjective quality. The starting point to accomplish this goal is the definition of models which quantify the HVS sensitivity to coding distortion. More precisely, these models link the HVS sensitivity to image features or compression tools (e.g., the selected frequency decomposition method, usually the DCT). This link can be found either by experimental observations or by explicit modeling of the HVS behavior. An example of experimental observation is the well-known phenomenon whereby the HVS is less sensitive to coding artifacts introduced in textured and very busy image areas [8]. From this observation a nonlinear relationship between the energy of the pixels in each

image area and the distortion perceived in that area can be devised. Conversely, an explicit modeling of the **eye behavior** is related to the sampling of an observed image performed by the retina with its rods and cones photoreceptors [8]. This sampling introduces a loss of details which can be modeled as a parameterized band-pass filter. The filter parameters can be then found by subjective experiments where observers rate the visibility of different amounts of noise introduced at different spatial frequencies.

Once a model for the HVS sensitivity to distortion is available, different tools for the aforementioned coding modules can be designed. As an example by considering the quantization stage, the step to discard and/or scale the frequency coefficients can be now adjusted to keep reconstruction error below the HVS sensitivity threshold. In [Section 5.05.4](#) several practical examples of use of HVS sensitivity models in the design of perceptual coding tools will be presented and discussed.

In this chapter, the sensitivity to coding artifacts is quantified by means of the ***Just Noticeable Distortion (JND)*** concept. JND refers to the minimum visibility threshold below which no change can be perceived. Alternatively, the JND can be seen as the maximum amount of distortion that can be introduced in a given image without being noticeable by any observer. The JND is a property of images and relates to certain masking phenomena of the HVS. The modeling and the details of the JND related to the HVS masking phenomena are the topics of the following section. At this point a simple example is given to explain the JND concept and to show how it can be measured. Consider the example of subjective quality evaluation from [Figure 5.4](#) whereby the horizontal axis denotes the level of distortion (e.g., coding artifacts) introduced in the original image. The starting point is the distortion level zero which corresponds to the original visual stimulus. Then, gradually a distortion  $D_i (i = 1, \dots, N)$  is added to the original image and a pool of observers are asked whether they can notice any difference. If the majority of the subjects answer no, the process continues adding distortion until, for a distortion level  $D_k$  the majority of the observers notice a difference with respect to the original. The distortion  $D_k$  represents the just noticeable distortion level after which any impairment becomes visible. The JND measured in this very simple example does not take into account all the different human visual system characteristics which contribute to it. The next section will detail the HVS mechanisms which provide the overall JND and how they can be analytically modeled.



**FIGURE 5.4**

Illustrative example of just noticeable distortion measurement for one image.

### 5.05.3 The modeling of the human visual system sensitivity to coding artifacts

This section introduces the modeling of the human visual system sensitivity to errors in compressed video, measured in terms of just noticeable distortion. The HVS sensitivity depends on visual masking phenomena which can be classified as either spatial or temporal. Spatial masking is related to characteristics of still images and the domain where the JND is measured (i.e., pixel or transform domain). In the transform domain, the main spatial masking phenomena are related to spatial frequency content, luminance, and contrast. Temporal masking is related to the motion activity along video frames and the domain where the JND is measured (i.e., pixel or transform domain). In the following, these masking phenomena are detailed and the models proposed to measure their contributions to the HVS sensitivity to coding artifacts using the JND paradigm are presented. Moreover, the common approach to combine all the masking phenomena and the methodology to assess a model of the HVS sensitivity are also discussed. As stated in the Introduction, this chapter only addresses the transform domain as this is the domain where errors are introduced by quantization in video codecs belonging to the MPEG-x and H.26x families. For the HVS modeling in the pixel domain the interested reader is referred to [8, 19, 20].

#### 5.05.3.1 HVS frequency masking and the associated just noticeable distortion model

Frequency masking refers to the well-known property that the human visual system is highly sensitive to distortions added to low spatial frequency components. More precisely, the HVS has band-pass filter properties. The band-pass filter frequency response associated to frequency masking was first experimentally derived by Mannos and Sakrison [21] and then corroborated by the experiment described in [22]. The formulation for the frequency response is given by:

$$H(f_s) = (a + b \cdot f_s) \cdot e^{-c \cdot f_s}, \quad (5.1)$$

where  $f_s$  denotes the spatial frequency (in cycles/degree) and  $a$ ,  $b$ , and  $c$  are parameters whose values are derived by experimental data fitting. In the model (5.1),  $H(f_s)$  denotes the HVS sensitivity to distortion, often referred as to contrast sensitivity, which is the capability of the HVS to distinguish different visual stimuli. From this definition it can be observed that the contrast sensitivity values are proportional to the reciprocal of the just noticeable distortion values. In fact, for image areas where different stimuli are easy to distinguish (i.e., high contrast sensitivity) the just noticeable distortion is very low and vice versa. Therefore, the JND associated with frequency masking is given in the literature as the reciprocal of the contrast sensitivity  $H(f_s)$ . The experimental data employed to derive the contrast sensitivity function (5.1) are usually collected in a subjective experiment with a number of observers by performing following steps:

1. Select a given spatial frequency  $f'_s$  and generate a sinusoidal grating at this particular frequency.
2. Superimpose the sinusoidal grating to an image with all pixel values at the mid gray luminance intensity to obtain the test image  $I_{test}$ .
3. Consider a zero mean random noise with intensity  $\eta$  that equals the minimum noise intensity allowed for this experiment (where this minimum noise intensity is derived by preliminary subjective viewings).

4. Add the noise  $\eta$  to the image  $I_{test}$  to obtain the noisy image  $I_{noisy}$ . Ask the subjects whether they can see any difference between  $I_{test}$  and  $I_{noisy}$ .
5. If a percentage of the subjects can see a change between  $I_{test}$  and  $I_{noisy}$ , then set the current value of noise  $\eta$  as the JND level for the frequency  $f'_s(\text{JND}(f'_s))$  and go to Step 1 to evaluate another spatial frequency.
6. Otherwise, increment the noise intensity  $\eta$  by the amount  $\Delta_\eta (\eta \leftarrow \eta + \Delta_\eta)$  and go to Step 4.

The model in (5.1) has a general formulation which holds for any frequency decomposition (i.e., transform). However, as stated in the Introduction, this chapter considers video codecs which perform quantization using the Discrete Cosine Transform (DCT). Therefore, formula (5.1) should be extended to account for the DCT. In particular the first extension concerns the spatial frequency  $f_s(i, j)$  in the 2-D DCT domain:

$$f_s(i, j) = \frac{1}{2 \cdot B} \sqrt{(i/\vartheta^x)^2 + (j/\vartheta^y)^2}, \quad (5.2)$$

where  $B \times B$  is a block of DCT coefficients, and the indexes  $(i, j)$  assume values in the range  $(0, \dots, B-1)$ . Finally,  $\vartheta^x$  and  $\vartheta^y$  are the horizontal and vertical visual angles of a pixel which can be obtained as [23]:

$$\vartheta^{x,y} = 2 \cdot \arctan\left(\frac{1}{2 \cdot R_{vd} \cdot H}\right), \quad (5.3)$$

where  $R_{vd}$  denotes the ratio between the viewing distance and the picture height  $H$ . In Eq. (5.3) the visual angles  $\vartheta^x$  and  $\vartheta^y$  have the same value because the Pixel Aspect Ratio (PAR) in modern monitors can be considered equal to one [23].

The model for contrast sensitivity in (5.1) does not take into account two other important properties of the HVS: (1) *directionality* which leads the HVS to be more sensitive to distortions added to horizontal and vertical frequencies rather than to diagonal ones and (2) *distortion summation* which increases the HVS sensitivity to distortion for a given range of spatial frequencies. These two properties have been modeled by Ahumada and Peterson in [24] as corrective terms for the model in (5.1), leading to the final form of JND associated with the frequency masking ( $\text{JND}_{FM}$ ):

$$\text{JND}_{FM}(i, j) = s \cdot \frac{1}{\phi_i \cdot \phi_j} \cdot \frac{e^{c \cdot f_s(i, j)} / (a + b \cdot f_s(i, j))}{r + (1 - r) \cdot \cos^2(\varphi(i, j))}, \quad (5.4)$$

where the term  $s$  takes into account the *distortion summation* effect,  $r$  accounts for the *directionality* effect, and  $\varphi(i, j)$  is the directional angle corresponding to the  $(i, j)$  DCT frequency component:

$$\varphi(i, j) = \arcsin\left(\frac{2 \cdot f_s(i, 0) \cdot f_s(0, j)}{f_s^2(i, j)}\right). \quad (5.5)$$

Finally,  $\phi_i$  and  $\phi_j$  in (5.4) correspond to the DCT normalization coefficients where, for both horizontal and vertical frequencies the following applies:

$$\phi_i = \begin{cases} \sqrt{1/B}, & \text{if } i = 0, \\ \sqrt{2/B}, & \text{otherwise,} \end{cases} \quad (5.6)$$

where  $B$  is still the DCT block size. The work in [23] suggests using values  $s = 0.25$  and  $r = 0.7$  in (5.4). Moreover, [23] also derives the values for parameters  $a$ ,  $b$ , and  $c$  according to an experiment performed in the same fashion as in Steps 1–6 which considers an  $8 \times 8$  floating point DCT. The values for these parameters are:  $a = 1.33$ ,  $b = 0.11$ , and  $c = 0.18$ . Finally, interested readers are referred

to [25] for the modeling of the  $JND_{FM}$  term when the wavelet transform is used as the frequency decomposition tool.

### 5.05.3.2 HVS luminance masking and the associated just noticeable distortion model

Luminance masking relates to the lower sensitivity of the HVS to the distortion introduced in darker and brighter image areas. This masking phenomenon derives from the Weber-Fechner law which states that the minimum perceivable visual stimulus difference increases with the background luminance. A typical example of this phenomenon is the difficulty of seeing a light bulb when turned on if the background is very bright. Moreover, given the ambient lighting surrounding the display, even the noise added to very dark image areas becomes less visible than the same amount of noise added to mid-gray areas [19]. The overall effect of luminance masking on the HVS sensitivity to distortion is a U-shaped curve which defines the maximum amount of distortion tolerated for considered luminance values. This curve is often referred to as a luminance masking JND profile ( $JND_{LM}$ ) and its values can be derived according to the subjective experiment described by the following sequence of steps.

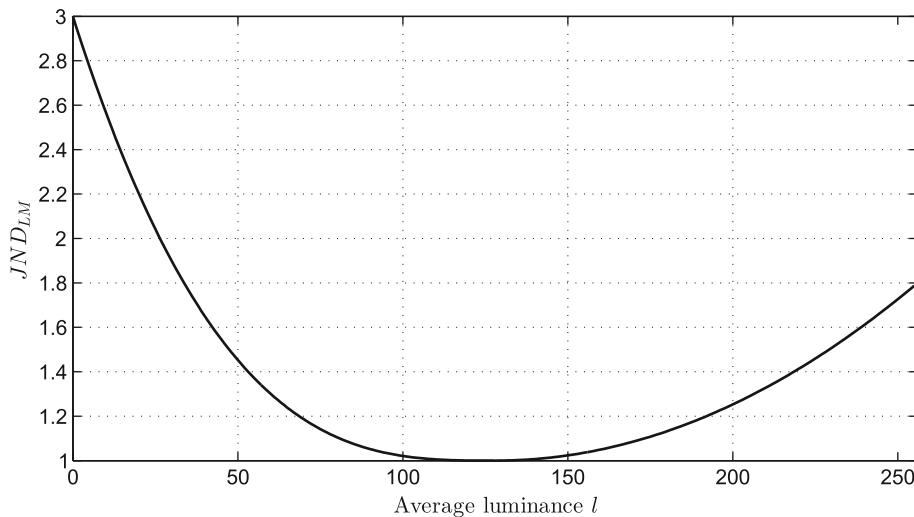
1. Consider the image  $I$  and set the initial luminance value  $l = 0$ .
2. Set all the pixels in  $I$  to the luminance value  $l$ .
3. Add a zero mean random noise with intensity  $\eta$  to an  $M \times M$  pixels area located at the center of  $I$ .
4. If a given percentage  $\alpha$  of the subjects can see the intensity difference related to the noise addition, set  $\eta$  as the JND for the luminance level  $l$  ( $JND_{LM}(l)$ ). Set  $l = l + \Delta_l$  and go to Step 2.
5. Otherwise, increment the noise intensity  $\eta$  by the amount  $\Delta_\eta$  ( $\eta \leftarrow \eta + \Delta_\eta$ ) and go to Step 3.

According to [19], the value for  $M$  can be set to 32. Generally, the initial intensity value for the random noise ( $\eta_{ini}$ ), the increment  $\Delta_\eta$ , and the background luminance  $\Delta_l$  increment are set according to some preliminary experiments. In particular, these experiments aim at finding a good set of values for  $(\eta_{ini}, \Delta_\eta, \Delta_l)$  to mitigate the trade-off between experiment accuracy and experiment length. In fact, suppose for example that for  $N$ -bit pixel precision the following values are set:  $\eta_{ini} = 0$ ,  $\Delta_\eta = \Delta_l = 1$ . Instead of performing exhaustive experiments over all possible intensity values ( $2^N$ ), in practice the evaluation can be done on a subset of those values with values for  $\Delta_l$  equal to 2 or 3 if  $N = 8$  is considered. The JND values for luminance intensities that have not been evaluated are interpolated by fitting a curve over the collected data. The work in [23] proposes to use a linear piecewise polynomial which, assuming 8-bit pixel precision, is defined as:

$$JND_{LM}(l) = \begin{cases} (60 - l) / 150 + 1 & l \leq 60, \\ 1 & 60 < l < 170, \\ (l - 170) / 425 + 1 & l \geq 170. \end{cases} \quad (5.7)$$

Conversely, the work in [26] fits a higher degree polynomial given as follows, again for 8-bit pixel precision:

$$JND_{LM}(l) = \begin{cases} 2 \cdot \left(1 - \frac{2 \cdot l}{256}\right)^3 + 1 & l \leq 128, \\ 0.8 \cdot \left(\frac{2 \cdot l}{256} - 1\right)^2 + 1 & \text{otherwise.} \end{cases} \quad (5.8)$$

**FIGURE 5.5**

JND profile for luminance masking defined as a higher degree polynomial proposed in [26].

Usually, the approximations in (5.7) or (5.8) are used over blocks in an image, i.e.,  $l$  is computed individually for each block. In the following, the luminance level for the  $k$ th image block will be denoted as  $l(k)$  and is assumed as the average of all the luminance values for the pixels in the block  $k$ . As a final example, Figure 5.5 shows the JND profile defined by (5.8).

### 5.05.3.3 HVS contrast masking and the associated just noticeable distortion model

Contrast masking describes the HVS property where coding artifacts are less noticeable in very busy and complex image areas rich in texture details. Such details tend to mask coding artifacts and therefore such properties of textures can be exploited to reduce the coding rate while maintaining the same perceptual quality. It is therefore necessary to understand how the HVS sensitivity responds to properties in these image areas. This property can be characterized by a relationship between the degree of texture detail for a given image area and the HVS sensitivity to the distortion added in this area. Foley and Boynton in [27] measured how the HVS sensitivity was influenced by random noise added over sinusoidal and Gabor patterns. In this experiment, the patterns used act as maskers, i.e., they represent the image texture, while the random noise represents the masking signal. This study found a nonlinear relationship between the JND level and the normalized masking energy. More specifically, the normalized masking energy  $E_n$  is defined as the ratio between the energy of the masker signal  $E_s$  (i.e., the image) and the energy of the masking signal which, for the considered lossy video coding scenario, corresponds to the coding artifact or quantization error  $E_q$ :

$$E_n = \frac{E_s}{E_q}. \quad (5.9)$$

For the case of the DCT, the normalized masking energy can be measured as proposed in [28]. In that work  $E_s$  is the square of the coefficient values as the DCT is an orthonormal transform.  $E_q$  is the square of the JND measured for that image. In fact, as will be further detailed in Section 5.05.4.1, the reconstruction error, i.e., the difference between the original and coded signal, is not perceivable when it is smaller than or equal to the JND. Since the model has to provide the maximum distortion level to be introduced without being noticeable, the reconstruction error is set to its upper bound, that is the JND associated to the frequency and luminance masking (since the contrast masking is to be determined here). Therefore, for the  $(i,j)$  DCT frequency component in the  $k$ th image block, the normalized masking energy  $E_n$  is defined as [28]:

$$E_n(i, j, k) = \left| \frac{C(i, j, k)}{\text{JND}_{FM}(i, j) \cdot \text{JND}_{LM}(l(k))} \right|^2, \quad (5.10)$$

where  $C(i, j, k)$  denotes the DCT coefficient  $(i, j)$  in block  $k$ . Once the normalized masking energy has been defined, the Foley-Boynton nonlinear relationship [27] for the JND associated to the contrast masking ( $\text{JND}_{CM}$ ) can be then expressed as:

$$\text{JND}_{CM}(i, j, k) = \begin{cases} 1 & \text{if } i = j = 0, \\ \max(1, E_n(i, j, k)^\varepsilon) & \text{otherwise,} \end{cases} \quad (5.11)$$

where the exponent  $\varepsilon$  is usually set to 0.6 as suggested in [28]. The exponent  $\varepsilon$  is referred to also as masking slope [29]. The  $\text{JND}_{CM}$  in (5.11) considers only the so-called intra-band masking effect, i.e., the masking due to the signal component within its own  $(i, j)$  DCT frequency. However, the contrast masking phenomenon involves also the inter-band masking [8,26] which is the equivalent of the inter-band masking for audio signals, whereby each frequency band can mask other bands. Several methods have been devised for modeling the inter-band masking contributions. The work in [29] proposes modulating the  $\varepsilon$  exponent in (5.11) according to the energy of the spatio-temporal gradient computed over the inter-frame difference between temporally adjacent frames. With this modification that takes into account the motion occurring between two consecutive frames, the  $\text{JND}_{CM}$  term is not only adaptive to spatial, but also to temporal characteristics of a video. As further clarified in the next section, such adaptation takes into account the temporal masking phenomenon. On the other hand, the works in [23,26] proposed to weight differently each frequency component using the  $\text{JND}_{CM}$  term depending on the image block type. More precisely, in [23] the image blocks are classified into plane, edge, or texture based on the number of contained edge pixels. Conversely, in [26] each block is classified into low, medium, or high masking depending on the sum of the absolute DCT coefficient values in different frequency groups. Depending on the image block classification and on the DCT component, different values are specified for  $\text{JND}_{CM}$ . For the  $\text{JND}_{CM}$  term, the adaptive masking slope weighting proposed in [29], the original Foley-Boynton nonlinear relationship depicted in formula (5.11) and the adaptive weighting based on block classification [23] have been compared in the work [30]. In particular all the three aforementioned models for the  $\text{JND}_{CM}$  term have been separately integrated in a H.264/AVC-based video codec to measure the associated bitrate reductions and subjective quality. The rate and the quality are compared against those of an H.264/AVC codec in the High profile. Good bitrate reductions for all the three models are reported with the adaptive masking slope weighting and the Foley-Boynton model providing the best results and the best detail preservation. Finally, it should be mentioned that inter-band masking effects

are more complex and are still to be fully understood [8]. Therefore, while a simple weighting for the  $JND_{CM}$  may work in practice, more studies are needed to fully address the inter-band masking effect.

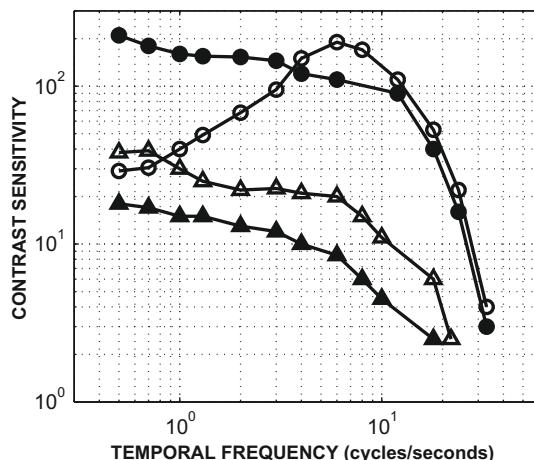
#### 5.05.3.4 HVS temporal masking and the associated just noticeable distortion model

The temporal masking of the HVS is associated with the motion in video sequences. The rationale behind the temporal masking is that the HVS sensitivity to coding artifacts is lower in areas with very high motion activity. Modeling temporal masking is more challenging because the spatio-temporal sensitivity function of the HVS is not separable ([12, 31]), i.e., it depends on both the spatial and temporal frequencies. Robson measured the temporal (contrast) sensitivity function of the HVS [32] and showed that for high temporal frequencies (i.e., greater than 10 Hz) the slope of the contrast sensitivity curve is constant for spatial frequency values either lower than 5 cycles per degree (cpd) or higher than 5 cpd. This function is demonstrated in Figure 5.6. In particular, by plotting the curves in the logarithmic domain, the slope at high temporal frequencies is set to  $-0.03$  [23, 33] leading to the following formulation:

$$CSF_T = 10^{-0.03 \cdot f_t}, \quad (5.12)$$

where  $CSF_T$  denotes the temporal Contrast Sensitivity Function (CSF) and  $f_t$  is the temporal frequency. The reciprocal of (5.12) gives the JND associated to the temporal masking ( $JND_T$ ) for high temporal frequencies (i.e.,  $>10$  Hz) in the following form:

$$JND_T = 1.07^{f_t} \text{ if } f_t > 10. \quad (5.13)$$



**FIGURE 5.6**

Temporal contrast sensitivity function obtained from the data given in [32]. Different curves refer to different spatial frequency values (cycle per degree, cpd):  $\circ = 0.5$  cpd,  $\bullet = 4$  cpd,  $\Delta = 16$ , and  $\blacktriangle = 22$  cpd.

For temporal frequencies lower than 10 Hz and spatial frequencies between 2 and 5 cpd, the temporal contrast sensitivity function is almost constant (see filled dot curve in Figure 5.6) with average value of 180 among all points. By normalizing the CSF function with respect to this average value and taking the reciprocal to get the  $JND_T$  term, it yields:

$$JND_T = 1 \quad \text{if } f_t < 10 \text{ and } 2 \leq f_s \leq 5. \quad (5.14)$$

Finally, the curve with open circles markers in Figure 5.6 (corresponding to 0.5 cpd spatial frequency) is not considered in state-of-the-art-models for the HVS sensitivity. In fact, given the usual tested viewing distances and current content spatial resolutions, the minimum spatial frequency value is 2 cpd [23]. Taking all these facts into account, the JND profile associated to the temporal masking is given as follows:

$$JND_T(i, j, n) = \begin{cases} 1 & \text{if } f_s(i, j) < 5 \text{ and } f_t(i, j, n) < 10, \\ 1.07^{f_t(i, j, n)-10} & \text{if } f_s(i, j) < 5 \text{ and } f_t(i, j, n) \geq 10, \\ 1.07^{f_t(i, j, n)} & \text{if } f_s(i, j) \geq 5, \end{cases} \quad (5.15)$$

where  $n$  denotes the  $n$ th frame in the video sequence. As mentioned above, the temporal CSF function of the HVS is not separable and depends also on the DCT spatial frequency  $f_s(i, j)$ . This dependency is highlighted in (5.15) for the temporal frequency  $f_t$ . To compute the temporal frequency  $f_t$  associated to  $f_s(i, j)$ , the motion in a video sequence is considered to be purely translational, with velocities along the  $x$  and  $y$  axes equal to  $v^x$  and  $v^y$ , respectively. This assumption is made to simplify the temporal frequency calculation. Consider a three-dimensional continuous time video signal  $s(x, y, t)$ . The Fourier transform,  $S(f_x, f_y, f_t)$ , for  $s(x, y, t)$  is given by:

$$S(f_x, f_y, f_t) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x, y, t) \cdot e^{-j \cdot 2\pi \cdot f_x^x \cdot x} \cdot e^{-j \cdot 2\pi \cdot f_y^y \cdot y} \cdot e^{-j \cdot 2\pi \cdot f_t \cdot t} dx \cdot dy \cdot dt, \quad (5.16)$$

where  $f_x^x$  and  $f_y^y$  denote the spatial frequencies along the  $x$  and  $y$  axes. Under the hypothesis of pure translational motion, each intensity stimulus at  $(x, y)$  along the time can be written with respect to the signal at time  $t = 0$ :

$$s(x, y, t) = s(x + v^x \cdot t, y + v^y \cdot t, 0). \quad (5.17)$$

Rewriting (5.16) under the pure translational motion assumption in (5.17) yields:

$$\begin{aligned} S(f_x, f_y, f_t) = & \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x + v^x \cdot t, y + v^y \cdot t, 0) \cdot e^{-j \cdot 2\pi \cdot f_x^x \cdot x} \\ & \cdot e^{-j \cdot 2\pi \cdot f_y^y \cdot y} \cdot e^{-j \cdot 2\pi \cdot f_t \cdot t} dx \cdot dy \cdot dt. \end{aligned} \quad (5.18)$$

By substituting,  $\alpha = x + v^x \cdot t$  and  $\beta = y + v^y \cdot t$  which imply  $d\alpha = dx$  and  $d\beta = dy$ , Eq. (5.18) can be rewritten as:

$$\begin{aligned} S(f_x, f_y, f_t) = & \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(\alpha, \beta, 0) \cdot e^{-j \cdot 2\pi \cdot f_x^x \cdot (\alpha - v^x \cdot t)} \\ & \cdot e^{-j \cdot 2\pi \cdot f_y^y \cdot (\beta - v^y \cdot t)} \cdot e^{-j \cdot 2\pi \cdot f_t \cdot t} d\alpha \cdot d\beta \cdot dt. \end{aligned} \quad (5.19)$$

After a little algebra, (5.19) becomes:

$$S(f_x, f_y, f_t) = \underbrace{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(\alpha, \beta) \cdot e^{-j \cdot 2\pi \cdot f_s^x \cdot \alpha} \cdot e^{-j \cdot 2\pi \cdot f_s^y \cdot \beta} \cdot d\alpha \cdot d\beta}_{S(f_x, f_y)} \cdot \int_{-\infty}^{+\infty} e^{-j \cdot 2\pi \cdot (f_t - f_s^x \cdot v^x - f_s^y \cdot v^y) \cdot t} \cdot dt. \quad (5.20)$$

this finally leads to the three-dimensional Fourier transform for a continuous time video signal:

$$S(f_x, f_y, f_t) = S(f_x, f_y) \cdot \delta(f_t - f_s^x \cdot v^x - f_s^y \cdot v^y), \quad (5.21)$$

where  $\delta$  denotes the Dirac's delta function. From (5.21) it may be noted that the three-dimensional Fourier transform of a video signal  $s(x, y, t)$ , with all pixels in a frame moving by translational motion only, is given by the two-dimensional Fourier transform  $S(f_x, f_y)$  of the video frame  $s(x, y)$ , sampled by the Dirac's delta function over the region of points in the  $(f_s^x, f_s^y, f_t)$  space which satisfy the following relation:

$$f_t = f_s^x \cdot v^x + f_s^y \cdot v^y. \quad (5.22)$$

Since this chapter focuses on video codecs which perform quantization in the DCT domain, the spatial frequencies  $f_s^x$  and  $f_s^y$  can be related to the DCT frequency at  $(i, j)$  positions for a  $B \times B$  block as follows:

$$f_s^x = \frac{i}{2 \cdot B \cdot \vartheta^x}, \quad f_s^y = \frac{j}{2 \cdot B \cdot \vartheta^y}, \quad (5.23)$$

where  $\vartheta^x$  and  $\vartheta^y$  are the horizontal and vertical visual angles of a pixel (see Section 5.05.3.1). Equation (5.22) identifies a plane in the  $(f_s^x, f_s^y, f_t)$  space and provides the formula to compute the temporal frequency for each spatial DCT component in the  $JND_T$  term from (5.15). The velocities  $v^x$  and  $v^y$  in (5.22) refer to the trajectories, along the  $x$  and  $y$  axes, of moving objects on the retina plane. It should be noted that if the human eye cannot track moving objects, their velocity on the retina plane would be the same as the one in the observed scene. However, the human eye is able to track moving objects by performing the so-called Smooth Pursuit Eye Movement (SPEM) which compensates the motion of objects on the retina plane and therefore has the general effect of reducing the object velocity on the retina plane as follows:

$$v^{x,y} = v_I^{x,y} - v_e^{x,y}, \quad (5.24)$$

where  $v_I$  and  $v_e$  denote the velocities on the image plane and the velocity of the eye movements, respectively. The velocity of one object moving on the image plane is computed still assuming pure translational motion. Therefore let  $d = (d^x, d^y)$  be the object displacement in pixel units between two temporally adjacent frames. It should be noted that this displacement corresponds to the motion vector usually computed for inter prediction in video codecs by performing block-based motion estimation. In order to compute the corresponding displacement in units of viewing distance, the visual angle  $\vartheta = (\vartheta^x, \vartheta^y)$  is taken into account. Finally, the velocity is obtained taking into account the frame rate as:

$$v_I^{x,y} = f_r \cdot \vartheta^{x,y} \cdot d^{x,y}, \quad (5.25)$$

where  $f_r$  denotes the frame rate in frames per second. The eye movement velocity ( $v_e$ ) is the result of two main contributions: natural drift eye movement and saccadic eye movement. Daly in [31] has proposed a model for the  $v_e$  velocity based on measurements for the two aforementioned eye movements

(i.e., drift and saccadic):

$$v_e^{x,y} = \min(g \cdot v_I^{x,y} + v_{drift}, v_{saccadic}), \quad (5.26)$$

where  $g$  is the object velocity reduction gain due to the SPEM which has typical value of 0.98,  $v_{drift}$  is the eye velocity due to the drift movement (typically  $0.15^\circ/\text{s}$ ), and finally  $v_{saccadic}$  is the eye velocity due to saccadic movements (typically  $80^\circ/\text{s}$ ). Finally, from (5.24) it can be seen that the velocity needed to compute the temporal frequency in (5.22) depends on the motion vector  $d = (d^x, d^y)$  which is associated to each block from the image. Therefore, the JND associated to the temporal masking ( $JND_T$ ) will hereafter be specified for each block. In the following section, all the JND terms related to the discussed HVS masking phenomena will be combined together in a complete spatio-temporal JND profile for the spatial and temporal masking of the HVS.

### 5.05.3.5 Combining all the masking phenomena in a spatio-temporal just noticeable distortion model

In order to combine the different HVS masking phenomena discussed above, approaches reported in the literature usually take into account all the different JND models by multiplying their values together to obtain the final JND for the spatial frequency component  $(i,j)$  in the  $k$ th block of frame  $n$ . The multiplicative approach has been confirmed by experimental evidence. More precisely, it has been found that frequency masking also depends on the luminance background intensity, so it would have also been influenced by the luminance masking. However, the band-pass frequency response in (5.1) has a similar shape for all the luminance background intensities [8]. What changes in (5.1) are the values' amplitudes which have been found to be proportional to the  $JND_{LM}$  term [8]. Therefore the model in (5.1) can be used with the  $JND_{LM}$  as corrective a term. The final spatio-temporal JND profile ( $JND_{ST}$ ) is given as:

$$JND_{ST} = JND_{FM} \cdot JND_{LM} \cdot JND_{CM} \cdot JND_T. \quad (5.27)$$

Literature on the modeling of the HVS masking phenomena is mainly related to the luma component of image and video. The chroma component has received less attention for two main reasons: first the HVS is more sensitive to the distortion introduced in the luma [8,9] and second because the chroma is differently defined according to the color spaces used. Often the subsampled chroma format (e.g., 4:2:0) typically does not account for the main bitrate burden and therefore is less important when significant bitrate reduction is targeted. However, for high fidelity and professional studio applications where high quality is sought, new studies toward the definition of JND models for chroma are needed.

### 5.05.3.6 Assessing a just noticeable distortion model

The evaluation of a JND model is an important step in its design and allows to quantify how good a model is in shaping the distortion in areas and frequencies whereas the HVS is less sensitive. From the assessment results it is possible to understand whether all the considered masking phenomena have been properly addressed and which are the model aspects that can be improved. Therefore, this section presents a common and general assessment methodology for JND models. The whole assessment procedure consists of a random noise injection and a subjective experiment.

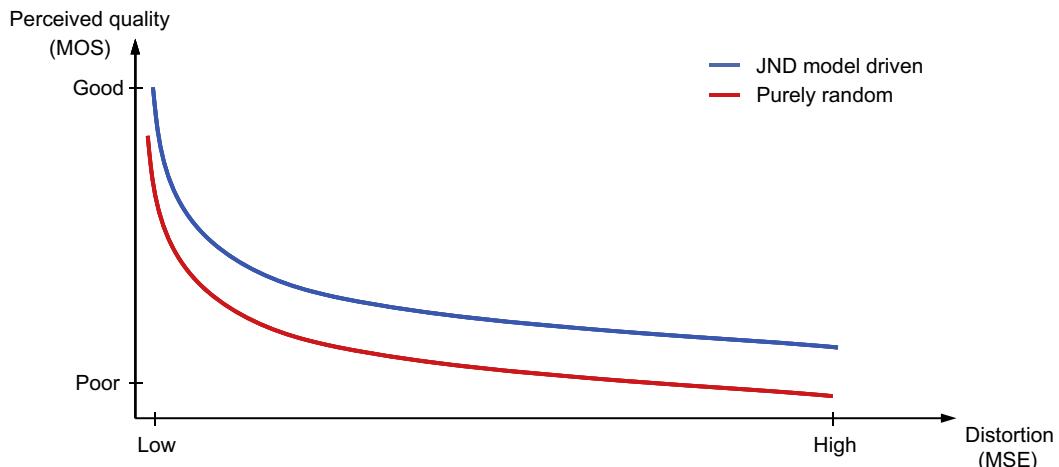
The random noise injection is done in two different modalities: *purely random* or *JND model driven*. In the purely random, a noise  $\xi$  is added to each frequency coefficient  $C$ . Conversely, in the JND model driven, the amount of noise  $\xi$  is controlled with the values provided by the JND model. For a given

image block, in transform domain the noise injection can be modeled as:

$$C' = C + g \cdot \xi \cdot \Phi, \Phi = \begin{cases} 1 & \text{if purely random,} \\ \text{JND}_{ST} & \text{if JND model driven,} \end{cases} \quad (5.28)$$

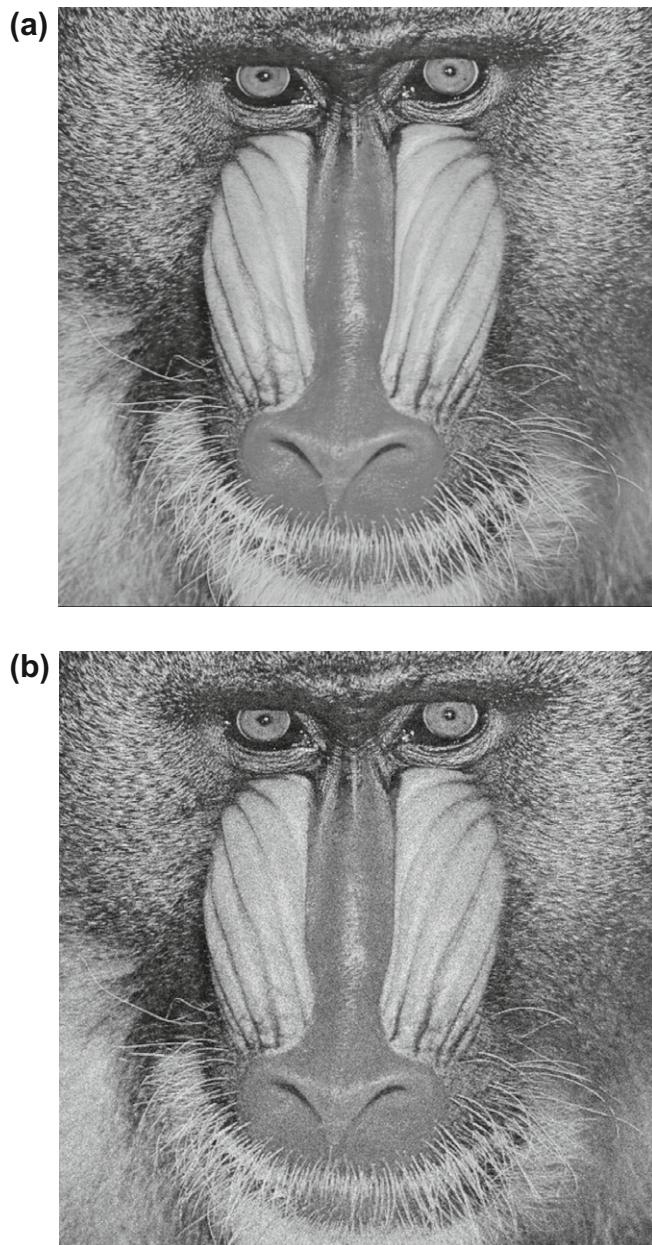
where  $C'$  is the coefficient corrupted by the noise  $\xi$  which randomly assumes values of  $\pm 1$  to avoid the introduction of fixed patterns and  $g$  is a gain factor which controls the amount of injected noise. The  $\Phi$  term in (5.28) assumes different values depending on the noise shaping modality (i.e., purely random or with JND model driven). It should be noted that a proper probability density function (pdf) of  $\xi$  may be selected to reflect the application domain where the JND model will be used. As an example, for quantization of DCT coefficients, a Laplacian pdf is usually selected [34].

In the subjective experiment, a pool of observers rates the quality of images or videos corrupted by noise injection according to the aforementioned modalities. In particular, for the same level of injected noise, the quality scores are collected and averaged to obtain the so-called Mean Opinion Score (MOS) [8, 13, 14]. Usually, the level of distortion is measured as the MSE with respect to the original content while the quality is rated on a continuous scale with values between 1 and 5 (i.e., poor to good quality). The MOS is collected for different distortion values and a distortion-quality curve is obtained as the one depicted in Figure 5.7. If the distortion-quality curve for the JND model driven noise injection is above the one related to purely random noise injection the evaluated JND model is confirmed subjectively. This means that, for the same amount of added distortion, the JND model is able to spread the noise in areas and frequency bands where the HVS is less sensitive. It should be noted that the obtained distortion-quality curves for different JND models can be also compared to evaluate which model hides noise better. In fact, the best model should have all the values of the distortion-quality curve greater than the other models considered for comparison. As a final example, Figure 5.8 shows the visual quality

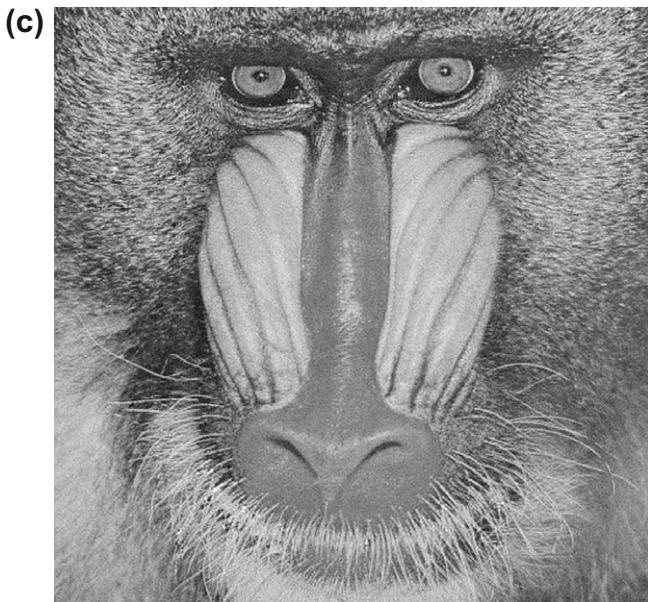


**FIGURE 5.7**

Example of distortion-quality curves obtained by assessing the quality of image or video content corrupted by *purely random* or *JND model driven* noise injection.

**FIGURE 5.8**

Comparison between purely random and JND model driven noise injection for the *Mandrill* image. (a) Luminance component for the original image. (b) Luminance component corrupted by purely random injection with MSE = 210. (c) Luminance component corrupted by JND model driven noise injection with MSE = 220.

**FIGURE 5.8**

(Continued)

obtained by noise injection with both purely random and JND model driven modalities. In particular, the same level of distortion (i.e., same MSE) is added to *Mandrill* test image. As may be noted, the region around the nose is better preserved in the JND model driven noise injection. In fact, the nasal septum region around the Mandrill's nose is smoother and more similar to the original in the JND model driven image rather than in the purely random noise injection image.

---

## 5.05.4 Integration of JND models in video coding architectures

After the review of the state-of-the-art for modeling the HVS sensitivity to coding artifacts given in [Section 5.05.3](#), this section addresses integration of the presented JND models in a video codec. As stated in [Section 5.05.2](#), once a model for the HVS sensitivity to distortion is available it can be used to design perceptual coding tools to be integrated in a video codec. Therefore this section discusses how a JND model can be integrated in the three coding modules mentioned in [Section 5.05.2](#): quantization, in-loop filter, and rate-distortion optimization. Moreover, the section also presents the normative functionalities provided by some state-of-the-art video coding standards to facilitate such integration.

### 5.05.4.1 Integrating a just noticeable distortion model in the quantization process

The integration of a JND model in the quantization process allows perceptual video coding to be performed by tuning the quantization step. To address how to link the thresholds provided by the JND

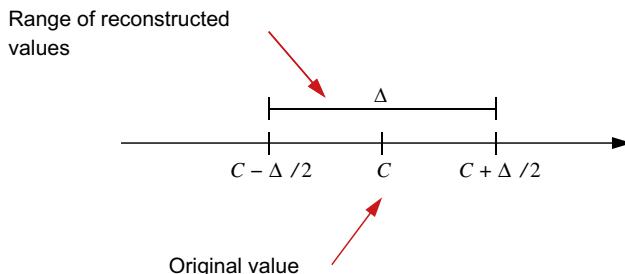
**FIGURE 5.9**

Illustration of the range of the reconstructed values after uniform quantization with quantization step  $\Delta$ .

model and the quantization step, consider a uniform quantizer with step  $\Delta$ . As depicted in Figure 5.9, when such a quantizer is applied to a given DCT coefficient  $C$ , the quantization error  $e_q$ , i.e., the absolute difference between the original and the reconstructed value, is always bounded in the range  $[-\Delta/2, \Delta/2]$ . The reconstruction error is below the just noticeable distortion if:

$$|C - \hat{C}| = |e_q| \leq \text{JND}, \quad (5.29)$$

where  $\hat{C}$  denotes the reconstructed DCT coefficient after quantization and JND is related to the frequency component considered, according to the JND models presented in Section 5.05.3. Provided that  $e_q$  is bounded in the range  $[-\Delta/2, \Delta/2]$  (see Figure 5.9), it yields:

$$\frac{\Delta}{2} = \text{JND} \Rightarrow \Delta = 2 \cdot \text{JND}. \quad (5.30)$$

Formula (5.30) shows the link between the quantization step variation for a given DCT coefficient and the amount of distortion which is tolerated for that coefficient (i.e., JND). Thus, to keep the video quality equal to the just noticeable distortion threshold, the quantization step  $\Delta$  for each DCT coefficient should not exceed twice the value of the associated JND threshold. Under these circumstances, the video quality obtained is *visually lossless* with respect to the original.

The condition in (5.30) specifies only one rate-distortion point which may be achievable only by a bitrate value that is not suitable for practical video coding applications. Therefore the quantization step should be increased to be higher than  $2 \cdot \text{JND}$  leading to the so-called *supra-threshold* condition whereby the coding artifacts are above the JND threshold. Under the *supra-threshold* condition it is not clear how the HVS sensitivity to distortion varies and different models are needed as pointed out in the early 1990s [9]. The current literature is lacking in practical proposals to tackle HVS sensitivity in the presence of visible coding artifacts.<sup>2</sup> However, the need to integrate HVS models in video codecs has led to the application of JND models in the supra-threshold condition assuming that the gap between two distortion visibility levels is always equal to the corresponding JND threshold. Under this assumption,

<sup>2</sup>To the best of the authors' knowledge, only the work in [35] addresses the problem for the wavelet transform and in a qualitative way.

the quantization step used for perceptual quantization (denoted as  $\Delta^{perc}$ ) is given by:

$$\Delta^{perc} = \Delta \cdot 2 \cdot \text{JND}, \quad (5.31)$$

where  $\Delta$  is the base quantization step and the constant 2 can be included in the JND thresholds.

#### 5.05.4.2 Integrating a just noticeable distortion model in the in-loop filter process

The integration of a JND model in the in-loop filter allows the tuning of the filter parameters to reduce coding artifacts. To this end, parameters such as the filter strength for deblocking can be modified to bring the coding distortion below the JND visibility threshold. More precisely, an iterative in-loop filter algorithm can be devised such that the filter parameters are varied under the allowed value range. After each filtering, the reconstructed error between the original and the filtered image area is measured and the process continues until this error is greater than the JND threshold. It should be noted that the described iterative perceptual filtering requires sending the parameters to the decoder. This can be done using standard functionalities (see [Section 5.05.4.4](#)) or by ad hoc techniques as proposed in [36].

#### 5.05.4.3 Integrating a just noticeable distortion model in the rate-distortion optimization process

One element of rate-distortion optimization is related to selection of the best coding mode for each coding block. The best coding mode is selected according to the following optimization problem:

$$\begin{cases} \min_m \{D_m\}, \\ \text{s.t. } R_m \leq R^*, \end{cases} \quad (5.32)$$

where  $m$  denotes the coding mode,  $D_m$  and  $R_m$  are the distortion and coding rates associated to  $m$ , and  $R^*$  is the rate budget for a given image block. The constraint problem in (5.32) can be reformulated as an unconstraint problem using the Lagrangian multiplier method [10]. In this case the cost function  $J$  is defined as:

$$J_m = D_m + \lambda \cdot R_m, \quad (5.33)$$

where  $\lambda$  denotes the Lagrangian multiplier. It should be noted that, by the JND definition, if  $D_m$  is below the JND visibility thresholds, it can be set to zero as viewers would not notice any difference with respect to the original. In this way, coding modes with same perceptual quality but lower coding rate are selected by the RDO and the overall compression efficiency is increased.

#### 5.05.4.4 Video coding standard functionalities to support integration of a just noticeable distortion model

This section discusses the functionalities offered by some widely used video coding standards (e.g., MPEG-2 and H.264/AVC) for the integration of HVS models. In this section, only normative functionalities, i.e., specified in the standard syntax, will be presented.

For the quantization process two main normative tools are available: quantization matrices and block level varying quantization step. Quantization matrices are tables of the same size of the transform blocks specified by a video coding standard and provide a means to vary the quantization step for each

coefficient. The matrices are space invariant, i.e., are the same for the whole picture, and standards like MPEG-2, H.264/AVC, and HEVC allow the transmitter to send different quantization matrices on a frame-by-frame basis. From the discussion in [Section 5.05.3](#), it can be recognized that quantization matrices account only for the HVS frequency masking and refer to the  $JND_{FM}$  term. By allowing the transmission of the matrix values, the encoder is able to select the matrices more suitable for a given video content. Usually there are different quantization matrices for each component type (luma and chroma) and frame coding type, i.e., intra or inter coding. For standards like H.264/AVC and HEVC whereby different block sizes are allowed for the transform, different quantization matrices are also used.<sup>3</sup> Block level quantization step variation is a tool specified in several video coding standards which enables both perceptual video coding as well as a finer step tuning for rate control purposes. Standards such as MPEG-2 and H.264/AVC allow variations in the quantization step for each  $16 \times 16$  block of luma pixels called a macroblock. Conversely, the emerging HEVC standard allows the step to be varied for block sizes ranging from  $64 \times 64$  to  $8 \times 8$ . The varying quantization steps for all the image blocks are differently encoded and written into the bitstream. By using the block level quantization step variation, the perceptual video quality can be improved by performing finer quantization in areas where the HVS is more sensitive to the distortion and a coarser quantization otherwise. The quantization step selection can be driven by different features such as the average JND threshold computed over the block or average luma or chroma pixels activity as in the MPEG-2 Test Model 5 (TM5) [37]. A TM5-like quantization step selection has been also implemented in the current reference software model for the HEVC standard (HM) [38]. From the discussion in [Section 5.05.3](#), it should be clear now that the block level quantization step variation can be used to account for the HVS luminance masking (i.e., the  $JND_{LM}$  term).

For the in-loop filter module, both the H.264/AVC and the new HEVC standard allow the filter parameters to be sent on a slice basis. The parameters that can be sent are offsets which control the amount of filtering for a given slice (i.e., a given set of macroblocks for H.264/AVC or a given set of Coding Tree Units (CTUs) for HEVC [18]). The value for these transmitted offsets can be selected to preserve the sharpness of small spatial details or to increase the amount of filtering in areas whereby blocking artifacts are very strong. The size of one coded slice and the filter offset are all degrees of freedom which can be adaptively selected by the encoder to improve the perceptual quality of the coded video. In particular, according to what is described in [Section 5.05.4.2](#), a perceptual metric based on a JND model can be used to optimize the filter parameters and slice size. More precisely, the perceptual distortion provided by the used metric and the bitrate spent to encode the filter parameters can be optimized using a Lagrangian cost function [10].

The aforementioned quantization matrices, block level quantization step selection and in-loop filter, may be used to build more perceptually tuned video codecs while still producing bitstreams compliant to the standard syntax. However, the price paid for this compliance is that fewer HVS masking phenomena can be considered (e.g., because only quantization matrices are used) and that the granularity at which the coding rate can be perceptually allocated is limited (e.g., the luminance masking is considered only over blocks larger than those where the DCT is applied). In the next section several perceptual video coding solutions will be reviewed and among them there will be some which require normative changes to the standard in order to perform perceptual coding. Their benefits against perceptual coding techniques which do not require any syntax change will be reported and discussed.

---

<sup>3</sup>In the HEVC standard the values for  $16 \times 16$  and  $32 \times 32$  matrices are derived by up-sampling from an  $8 \times 8$  matrix.

---

## 5.05.5 Practical perceptual video coding schemes

This section reviews the state-of-the-art Perceptual Video Codecs (PVCs) proposed in the literature. Firstly, solutions for integrating just noticeable distortion models into video codecs are discussed. Then, principles behind PVCs which do not explicitly integrate JND models in the coding process, while still exploiting the HVS properties to reduce the coding rate, are explained. Finally, benefits of employing a PVC with a low-complexity integration of a JND model are demonstrated in the challenging environment of a highly optimized HEVC codec.

### 5.05.5.1 Perceptual video codecs with integration of just noticeable distortion models

Work on PVCs during the last two decades was not only driven by advances in HVS modeling, but also by the feasibility of integrating those models into available video codecs. Standardized video codecs are imposing numerous constraints for integration of JND models, thus limiting perceptual coding capabilities. Despite this, numerous studies mentioned in the remainder of this subsection reveal the potential benefits of including perceptual tools in mainstream video codecs.

- *3D pixel domain subband PVC*: Chou and Chen in [39] proposed a pixel domain JND model which extended their spatial JND model proposed in [19] by considering the temporal masking of the HVS. The spatial JND model considers the luminance and contrast masking phenomena and was originally proposed for image coding with the JPEG standard. In this work instead, the JND associated to the temporal masking is represented by a U-shaped curve, similar to the one for luminance masking (see [Section 5.05.3.2](#)). The curve is a function of the luminance difference between temporally adjacent frames and for each difference value provides the corresponding JND threshold. For high difference values (high motion activity) the JND threshold is higher than low inter frame values (low motion activity). The spatial JND model and the U-shaped curve profile are combined according to the multiplicative approach described in [Section 5.05.3.5](#). The obtained spatio-temporal JND model is then integrated in a 3D wavelet codec to perceptually allocate the available source coding rate as well as to perform unequal error protection for robust transmission over error-prone channels. More precisely, the perceptual source coding rate allocation exploits the spatio-temporal JND model by allocating less bits to higher spatio-temporal frequencies using higher JND thresholds. In the same fashion, the channel redundancy is stronger for low spatio-temporal frequencies and where the JND thresholds are lower, i.e., in areas where the HVS sensitivity to error is higher. The comparison was done against equal error protection using both the Peak-Signal-to-Noise-Ratio (PSNR) and the Peak-Signal-to-Perceptible-Noise-Ratio (PSPNR, see [19] for its analytical definition). Experimental results show that performing unequal error protection driven by the designed JND model provides higher PSNR and PSPNR values for a bit error rate of  $10^{-2}$ .
- *Perceptual pre-processing of motion compensated residuals*: Yang et al. in [20] extended the pixel domain spatial JND model proposed in [19] by considering the application of JND to chroma components. In particular, the spatial JND model proposed by Chou and Chen is now computed over the luma and the two chroma components. For the chroma components the  $JND_{LM}$  for the luminance and the considered chroma component is combined according to a nonlinear function. This combination models the masking effects of one component with respect to the other (e.g., luma

with respect to chroma). The derived luma-chroma JND model has been firstly assessed according to the same methodology presented in [Section 5.05.3.6](#). In this assessment, subjective tests showed that the proposed model was able to provide the same perceptual quality of images and videos that were corrupted with purely random noise. In particular, at the same perceptual quality, PSNR was measured for different test material which showed that the content where the noise injection was JND model driven had up to 2 dB lower PSNR than the content corrupted with purely random noise injection. In other words, the proposed luma-chroma JND model can provide the same perceptual quality of purely random noise injection but tolerates more distortion. The proposed luma-chroma JND model has been then tested in an MPEG-2 codec. The model was used to *pre-process* the prediction residuals obtained by subtracting the original from the motion compensated predictor. In this context, pre-processing consists of setting to zero those residuals whose magnitude was lower than their corresponding JND threshold. In this way, the residual energy is smaller and thus the coding rate is reduced. Experiments over several test videos have demonstrated that this perceptual video codec increases the PSPNR up to 0.25 dB against an MPEG-2 codec at the same bitrate.

- *Perceptual video codec with adaptive masking slope:* Leung and Taubman [29] adopted the wavelet domain JND model proposed in [25] for application in video compression. This model considers all the aforementioned spatial masking phenomena (i.e., frequency, luminance, and contrast masking). The contrast masking term ( $JND_{CM}$ ) has been further improved in this work by making it adaptive with respect to the spatio-temporal video features. In particular the masking slope for  $JND_{CM}$  (i.e., the  $\varepsilon$  exponent in (5.11)) varies according to the magnitude of the spatio-temporal gradient measured over the difference of two temporally adjacent frames. The obtained JND model is then used to weight the distortion function (MSE) in the RDO process of a wavelet-based scalable video codec. More precisely, the reconstruction error is scaled by the obtained JND thresholds and therefore the distortion is perceptually weighted. The weighting results in a more frequent selection of coding modes which have lower coding rate and provide the same perceptual quality than other modes which have higher coding rates. The performance of this wavelet-based perceptual video codec is reported both in terms of objective and subjective measurements. For the objective assessment, the Perceptual Distortion Metric (PDM, [29]) has been used. This metric provides the values in a scale  $[0, +\infty)$  where the lower the value, the closer the coded content quality is with respect to the original. At the same bitrate, the proposed wavelet PVC achieves a reduction in average PDM values of 20%, with respect to the base wavelet codec used in this evaluation. For the subjective assessment, a pool of observers voted the perceptual quality of coded videos at the same bitrate. The MOS scores obtained with the proposed wavelet PVC and the ones obtained with the base wavelet codec are compared in terms of percentage quality improvement. At the same bitrate an average subjective quality improvement of about 20% is reported.
- *Perceptual suppression of transform coefficients:* Mak and Ngan in [40] used the DCT domain JND model proposed in [41]. This model considers all the spatial and temporal masking phenomena discussed in [Section 5.05.3](#). In particular, the contrast masking model considers both intra-and inter-band effects for the contrast masking phenomenon. All the models for the considered spatial and temporal masking phenomena are then combined according to the multiplicative approach described in [Section 5.05.3.5](#). The obtained spatio-temporal JND profile is then used to suppress (i.e., set to zero) the DCT coefficients of the prediction residuals with absolute values below the corresponding JND threshold. It should be noted that this coefficient suppression is the same as the motion compensated residuals pre-processing described previously and proposed in [20]: the main difference here regards

the domain where the residuals are set to zero. The adopted spatio-temporal JND model has been also used in the RDO to weight the MSE between the original and the coded block, similarly as proposed in the wavelet codec with adaptive masking slope previously reviewed. Using the JND suppression, the RDO selects coding modes which provide the same perceptual quality with lower bitrate. This approach was integrated in an H.264/AVC codec and its performance was subjectively evaluated for high definition videos. The results show that an average bitrate reduction of up to 23% can be achieved while maintaining the same perceptual quality, compared to the H.264/AVC codec.

- *Foveated pixel domain JND perceptual codec:* Chen and Guillemot in [42] considered the spatio-temporal pixel domain JND model previously reviewed for the 3D pixel domain subband PVC proposed by Chou and Chen in [39]. It has been further observed that the image areas which are more attractive to viewers increase the sensitivity to distortion. Therefore, they also considered the *foveation* model proposed in [43] to improve their adopted spatio-temporal JND model. The fovea denotes the pit in the retina which provides the maximum acuity of vision [8]. The high concentration of photoreceptor cells in the fovea is responsible for greater sensitivity of the HVS to distortion in image areas currently projected to the fovea. This effect can be modeled by providing a space variant weighting for different image areas, giving higher importance to areas that attract more attention. The obtained JND-Foveated model (*JNDF*) can be then used to drive quantization strength during video compression, such that areas with higher average JND threshold values are quantized more. In the proposed solution, the block level quantization step adjustment is used to vary the quantization step for different macroblocks in an H.264/AVC compliant video codec. More precisely, for each macroblock an average JND threshold is obtained from all the *JNDF* values associated to pixels in the macroblock. The average JND threshold is then used as input to a nonlinear mapping to obtain the final quantization step. Macroblock level quantization step adjustment is a functionality of H.264/AVC which was here used to transmit actual quantization steps to the decoder. The performance of the foveated JND application has been subjectively evaluated and compared with that of the H.264/AVC codec. At the same rate, the observers rated the subjective quality of the videos coded with the two considered codecs. The subjective quality is measured on a scale between 0 and 100 where higher scores are related to better quality. The proposed foveated PVC gave a MOS difference of up to 10 points against the H.264/AVC codec. It should be noted that a pixel level precision for rate allocation using the *JNDF* model can be achieved. However, in that case a pixel-precise weighting model would need to be transmitted to the decoder.
- *Fine perceptual rate allocation with transform domain JND model:* Naccari and Pereira in [44] proposed the so-called Instituto Superior Técnico-Perceptual Video Codec (IST-PVC). The IST-PVC design considered the DCT domain JND model proposed by Wei and Ngan in [23]. This spatio-temporal JND model accounts for all the spatial and temporal masking phenomena discussed in [Section 5.05.3](#) and combines all the masking models according to the multiplicative approach presented in [Section 5.05.3.5](#). Since this spatio-temporal JND model was integrated in an H.264/AVC codec, the range for the JND thresholds has been modified to account for the integer DCT used in the standard. The JND thresholds are then used to vary the quantization step for each DCT coefficient. In order to avoid the transmission of the varying quantization step for each coefficient to the decoder, a decoder side JND model estimation was devised, which uses the motion compensated predictor to estimate the JND thresholds. The IST-PVC has been compared with the H.264/AVC codec in the High profile and the foveated PVC from [42]. The comparison is reported in objective terms by means of two video quality metrics: the Video Quality Metric (VQM) and the MOTion-based Video

Integrity Evaluation (MOVIE) [14]. At the same objective quality values obtained by the H.264/AVC codec in the High profile, the IST-PVC can provide an average bitrate reduction of 30%. For the same setup, the foveated PVC provides an average bitrate reduction of 10%. The same decoder side JND model estimation has been also used to target the emerging HEVC video coding standard in the work [45]. The perceptual HEVC codec has been compared with the HEVC codec using the Multi Scale-Structural SIMilarity (MS-SSIM) as an objective quality metric. It is reported that the HEVC PVC can provide the same objective quality obtained by the HEVC codec at an average bitrate reduction of 16%. Further extension of this work has been recently done in the context of HEVC. The details and results on this extension are summarized in [Section 5.05.5.3](#).

### 5.05.5.2 Perceptual video codecs based on image analysis and completion

Alternatively to using perceptual video codecs which explicitly integrate and use models of the human visual system in the coding process, reduction of coding rate by considering HVS properties can also be achieved using Image Analysis and Completion-Video Coding (IAC-VC) [46].

The main idea behind the IAC approach is to perform *image analysis* to divide each picture in two main areas: *perceptually relevant* and *perceptually irrelevant*. The former areas are encoded with conventional block-based motion compensated techniques while the latter, characterized by high concentration of texture details, are not encoded but are synthesized at the decoder with perceptually oriented *completion* techniques.

This section briefly reviews the fundamental approaches in the area of IAC-VC, including classification of image completion methods and video codec designs that support IAC-VC.

#### 5.05.5.2.1 Common approaches for image completion

The term image completion denotes the substitution, concealment, inpainting, or synthesis of missing image pixels. There are three main approaches to perform image completion: parametric, Partial Differential Equation (PDE)-based, and non-parametric.

- *Parametric image completion:* The methods following this approach try to estimate the stochastic process underlying the image area being completed with some models with a finite number of parameters. Generally, the models used are Auto Regressive (AR) or Auto Regressive Moving Average (ARMA) predictors. The work in [47] divides each video frame into edge and non-edge blocks. Edge blocks are encoded with an H.264/AVC-based codec while the non-edge blocks are completed at the decoder by texture synthesis. The devised texture synthesis employs an AR model whose parameters are fitted over the non-edge blocks within a frame and transmitted to the decoder. High bitrate reductions (up to 54%) are reported with respect to the videos coded with the H.264/AVC at the same subjective quality. Conversely, the works in [48,49] use an ARMA model to complete texture in inter coded frames. The frame with texture completion is then inserted into the reference frame buffer and used for future motion compensated predictions. For the proposed approach, bitrate saving of up to 23% for the same subjective quality is reported.
- *Partial Differential Equation (PDE)-based image completion:* The methods following this approach formulate the image completion problem by means of a differential equation which imposes two constraints. The first constraint is related to the boundary ( $\Omega$ ) between the completed and the uncompleted region where the Dirichlet condition must hold. That is, the variation of the pixel

intensities inside the region to be completed must be same as the variation of those pixels along  $\Omega$ . Second, the prolongation of the pixels inside the region to be completed has to follow isophote lines, i.e., line of pixels with the same intensity value. The work in [50] proposes an image inpainting approach based on PDE for intra coding for the H.264/AVC standard. The PDE-based inpainting is added as an additional mode to the existing H.264/AVC intra prediction mode. It is reported that the adopted PDE-based image completion is well suited for spatial homogeneous image areas. When compared to the H.264/AVC standard in the Main profile this PDE-based image completion method can achieve a bitrate reduction of up to 3.5% for the same objective quality expressed as PSNR.

- *Non-parametric image completion:* The methods following this approach try to estimate the stochastic process related to the image area being completed by the available neighboring data [51]. Usually for each image pixel to be completed, a set of candidate pixels is tested in the image areas either already completed or decoded. For each candidate, a  $W \times W$  patch of surrounding pixels, including the candidate, is compared with the  $W \times W$  patch of pixels surrounding the pixel being completed. Between these two patches a distance measure is computed and the candidate which minimizes such distance is selected as the pixel for completion. Common distance metrics include the Mean Square Error (MSE) or the Sum of Absolute Difference (SAD). It should be noted that the non-parametric image completion can be seen as a sort of non-local means denoising algorithm [52] whereby the pixel being completed can be seen as the pixel to be filtered. Moreover, the non-parametric image completion approach can be further improved by using some prioritization techniques for the pixels to be completed. As an example, the approach proposed by Criminisi et al. [53] completes first those pixels surrounded by a high number of pixels with known values (i.e., already completed or decoded). Perceptual video codecs which use non-parametric image completion methods can be found in the works [50, 54]. In particular, the work in [54] proposes to average the candidates among different  $W \times W$  patches rather than using a single candidate for image completion.

Parametric methods are the least complex among the image completion approaches and they are well suited for prediction and reconstruction of smooth and non-structured areas. PDE-based methods are suitable for thin and elongated image regions [46] and their computational complexity is half way between parametric and non-parametric methods. Finally, non-parametric methods can efficiently reconstruct textured regions but can suffer from excessive computational complexity. Moreover, non-parametric methods can cause error propagation by completion from surrounding image pixels with reconstruction errors.

### 5.05.5.2.2 Practical image analysis and completion video coding schemes

The image completion techniques described in the previous section can be used in video coding architectures according to two approaches: block- and region-based schemes. Block-based schemes divide each video frame in two block classes: blocks to be coded with conventional methods and blocks not coded and completed at the decoder. One advantage of block-based IAC-VC is the reuse of image partitioning from the hosting codec. Examples of block-based IAC-VC schemes are the ones briefly described previously in the review of the common approaches for image completion and proposed in references [47–50, 54].

Conversely, in region-based IAC-VC schemes the video frame segmentation is region oriented and therefore there are regions which are coded with conventional methods and regions not coded and

completed at the decoder. The main advantage of region-based IAC-VC scheme is that the perceptual coding is optimized and adapted at the content level. In particular, background or image areas which do not attract the viewers' attention can be clustered together and synthesized at the decoder. However, the price to pay for region-based schemes is the increased complexity in the region segmentation for each video frame.

Although more focused on proposing a flexible representation for video content rather than designing a perceptual video codec, the work in [55] can be considered as one of the first region-based IAC-VC schemes. In this work, the input video is segmented into three maps: intensity, alpha channel, and motion, where each map is encoded separately. The main drawback of this method is a possibility to introduce visual artifacts by applying erroneous segmentation. Dumitraş and Haskell proposed a region-based IAC-VC scheme in [56] where the input video is divided into replicable and non-replicable regions. From the replicable regions, texture parameters are extracted and the texture is removed from the original video and then synthesized at the decoder. Non-replicable regions are instead encoded with an H.264/AVC codec. The evaluation over a selection of clips from movies showed that while preserving the same subjective quality as for the H.264/AVC codec, the proposed method provides bitrate reductions of up to 55%. The main drawback of this IAC-VC scheme is its poor performance for videos with fast local or global motion [46].

Zhang and Bull in [57] proposed a region-based IAC-VC scheme which employs texture warping and non-rigid texture synthesis as completion techniques. The whole video frame is segmented into three regions: rigid textures, non-rigid textures, and non-texture regions, which are then passed to the texture warping, non-rigid texture analyzer, and an H.264/AVC encoder, respectively. Over the reconstructed frame, the regions which have been synthesized with either texture warping or non-rigid texture synthesis are further passed to a visual quality assessment module which computes an objective metric to assess the quality of the synthesized textures. Those regions which present a low quality according to the quality assessment metric used are then encoded with an H.264/AVC codec. Compared to the videos coded with the H.264/AVC in the Main profile, the proposed method achieves up to 55% bitrate reduction at the same subjective quality.

Another recent region-based IAC-VC scheme is proposed by Ndjiki-Nya et al. in [58] where each video frame is segmented into homogeneous regions by means of a texture segmentation module. The parameters for texture synthesis are extracted from the homogeneous regions and the frame is synthesized assuming that all the textures are rigid textures. After this step, a video quality assessment module decides whether the region has sufficient quality. If not, the region is synthesized as non-rigid texture and its quality is evaluated again. If the quality is still not sufficient, H.264/AVC coding is performed over the region considered. This codec has been compared with a fast implementation of the H.264/AVC reference codec and a bitrate reduction of up to 41% is reported for the same subjective quality of the videos coded with the H.264/AVC standard.

### 5.05.5.3 Perceptual video codecs with integration of just noticeable distortion models and low complexity

The majority of perceptual video codecs proposed in the literature involve a significant amount of computational complexity. However, using simplifications and relying on the underlying codec's architecture, perceptual tools can be integrated in practical solutions without imposing a burden to either encoder or decoder. To demonstrate applicability of the perceptual tools, this section presents solutions for a

simplified integration of an extension of the luminance masking phenomena presented in [Section 5.05.3.2](#). The simplification is referring to a minimal burden this perceptual tool brings to a practical codec, achieved by reusing flexible coding concepts, integer-precision arithmetic, and without breaking practical processing pipelines used in a codec. To achieve this some trade-offs are needed, e.g., between fine granularity for perceptual rate allocation and standard compliance. However, as demonstrated at the end of this section, such compromises still lead to overall good performance of perceptual tools.

The JND model associated to the luminance masking has been presented in [Section 5.05.3.2](#). Related JND profiles  $JND_{LM}$  (e.g., from [\(5.7\)](#) and [\(5.8\)](#)) can be computed starting from the average luma value for each image block. The values for average luminance  $l$  can be limited to the values of luminance sample and computed by additions and bit shifting for averaging (in case of commonly used blocks whose number of pixels is a power of two). Limiting permitted values for  $l$  allows pre-computing  $JND_{LM}$  values for referencing during coding.

For standard compliance, the block quantization step adjustment may be adapted, to incorporate JND weighting. However, for standards like H.264/AVC and HEVC the minimum block on which the quantization step can be changed is larger than the prediction or transform block. Changing the quantization step size over blocks larger than the ones where transform is applied may not be beneficial for the perceived quality. There are two possible solutions to this problem: either modifying the standard syntax to allow the quantization step variation at the minimum block size used for transform or using decoder side estimations as the one proposed in [\[44\]](#). The syntax modification implies an increment in the bitrate spent to signal the varying quantization steps. Conversely, the decoder side estimation would avoid this additional bitrate by computing required quantization steps at the decoder side. Additionally, it would also allow use of a different quantization for the chroma component, so that one can use different JND profiles since, as stated in [Section 5.05.3.5](#), the chrominance requires further studies and different sensitivity models. It should be noted that in this case when decoder side estimation is used the standard compliance dimension is sacrificed since the estimation has to be specified in the standard. In particular, a decoder would have to support the quantization driven by the *intensity* of the pixels belonging to the image block. This concept is called Intensity Dependent Quantization (IDQ) and it has been evaluated for application in HEVC [\[59–61\]](#).

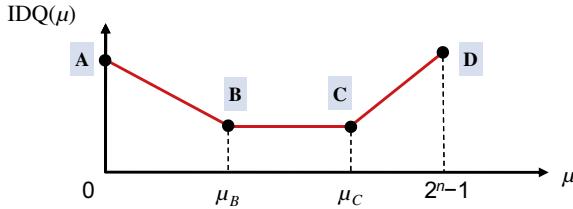
The IDQ perceptual coding tool extends the JND associated to luminance masking and uses the decoder side intensity estimation mechanism proposed in [\[44\]](#) to avoid the transmission of the varying quantization steps. The IDQ design has been driven by the following requirements:

- To enable flexibility in adapting to the picture characteristics.
- To enable low computational complexity implementation.

The following subsections will detail the design and implementation.

#### **5.05.5.3.1 Flexibility in adapting to the picture characteristics**

The IDQ tool performs perceptual coding by increasing the quantization step in image areas with higher and lower intensity values (i.e., darker and brighter image areas specified by  $JND_{LM}$ , [Figure 5.5](#)). Quantization step change can be varied on a predefined block type (e.g., transform block) and the variation consists of the multiplication by the base quantization step  $\Delta$  as shown in formula [\(5.31\)](#). Since the chroma components may present different characteristics with respect to the luma, the amount of quantization introduced on each color component can be different and the IDQ for chroma does not have

**FIGURE 5.10**

Parameterization of the IDQ profile with four characteristic points.

to follow  $JND_{LM}$ . Moreover, since the characteristics of video frames may change along the sequence, the transmitter may want to change the mapping between the average block intensity, hereafter denoted as  $\mu$ , and the quantization step; this mapping will be denoted as the IDQ profile. It should be noted that by allowing an encoder to define a different IDQ profile for each frame (or even for each coded slice) the amount of required bits due to the profile transmission may be significant, especially at low bitrates. To limit the amount of bits required by the profile, a profile curve can be parameterized. For example, as shown in Figure 5.10 where four characteristic points are highlighted, only the coordinates  $(\mu, IDQ(\mu))$  of the points A, B, C, and D are transmitted to the decoder. The whole IDQ profile is then recovered by interpolating the straight lines passing through the pairs (A, B), (B, C), and (C, D). Moreover, given that  $\mu_A \equiv 0$ ,  $\mu_B \equiv 2^{bd} - 1$  where  $bd$  denotes the pixel bit-depth and with additional simplification  $IDQ(\mu_C) \equiv IDQ(\mu_B)$ , the number of parameters sent to the decoder is further reduced, making the IDQ profile inexpensive for transmission.

While parameterization with four points only may be limited for some applications, sending full polynomial curves may be expensive. To balance the trade-off between the low bitrate for IDQ profile transmission and flexibility in the profile selection, the work in [60] proposes a different representation which is now able to accommodate different profile curves. The derivation of the new IDQ profile representation considers the relation between quantization step size ( $\Delta$ ) and the Quantization Parameter (QP), which is in HEVC approximately given as:

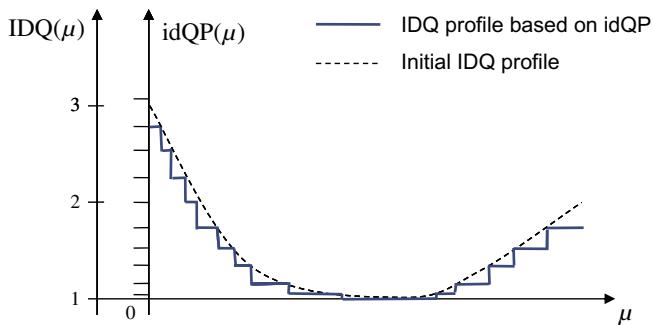
$$\Delta = 2^{\frac{QP-4}{6}}. \quad (5.34)$$

By recalling Eq. (5.31) to compute the perceptual quantization step size and inverting (5.34) the perceptual quantization parameter ( $QP^{perc}$ ), associated to perceptual quantization step ( $\Delta^{perc}$ ), can be computed. Moreover, by expressing  $QP^{perc}$  with respect to the baseline QP, the intensity differential Quantization Parameter (idQP) can be obtained:

$$\begin{aligned} QP^{perc}(\mu) &= 6 \cdot \underbrace{\log_2(\Delta \cdot IDQ(\mu))}_{\Delta^{perc}} + 4 = 6 \cdot \underbrace{\log_2(\Delta)}_{QP} + 6 \cdot \log_2(IDQ(\mu)) \Rightarrow \\ QP^{perc}(\mu) - QP &= 6 \cdot \underbrace{\log_2(IDQ(\mu))}_{idQP(\mu)}. \end{aligned} \quad (5.35)$$

Finally, in order to get only integer values for the QP and idQP quantities, it yields:

$$idQP(\mu) = \lfloor 6 \cdot \log_2(IDQ(\mu)) + 0.5 \rfloor, \quad (5.36)$$

**FIGURE 5.11**

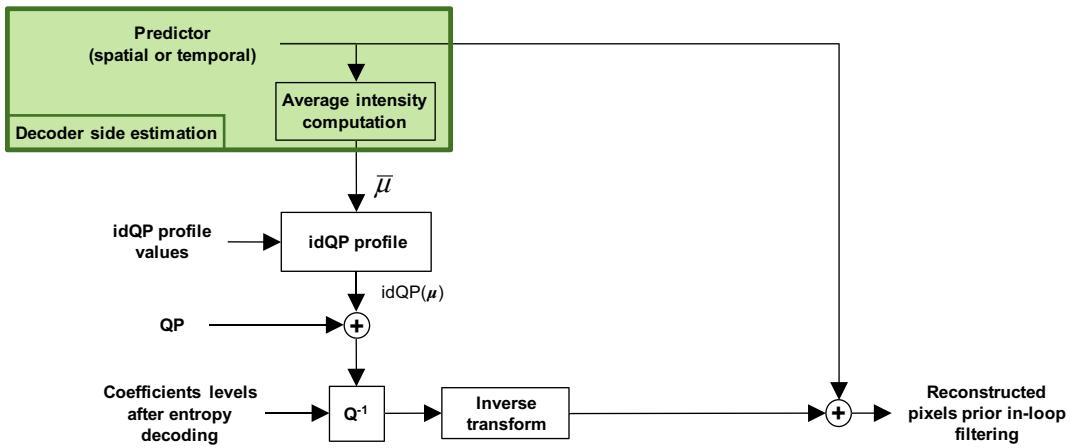
idQP profile representation compared with the corresponding IDQ profile.

where  $\lfloor \cdot \rfloor$  denotes the rounding to the nearest integer lower than the argument. The relation between the idQP and the  $IDQ(\mu)$  is also depicted in Figure 5.11 whereby the initial IDQ profile is translated to an idQP profile using Eq. (5.36). As shown in Figure 5.11 the IDQ profile represented in idQP units, assumes a staircase shape which can be efficiently signaled to the decoder. The idQP profile values can be encoded using Differential Pulse Code Modulation (DPCM) [60]. More precisely, the whole  $\mu$  range ( $[0, 2^n - 1]$ ) is divided into bins whereby each bin ( $b$ ) is characterized by its width ( $w$ ) and the quantity  $\delta = idQP(b_m) - idQP(b_{m-1})$ . An example is given in Figure 5.11 where idQP profile consists of 15 bins. While all bin values (i.e.,  $w$  and  $\delta$ ) have to be transmitted to the decoder, the required bitrate is lower than for transmitting all the profile values. Moreover, the presented simplified representation provides flexibility of the profile shape. Additional simplification can be introduced by limiting  $\delta$  to values of +1 or -1. This reflects properties of typically smooth variations of the IDQ profiles (e.g., [23, 26] presented in formulas (5.7) and (5.8)).

It should be noted that the conversion of the IDQ profile to the idQP units not only allows efficient transmission of generic curves used for IDQ profiles but also reduces the computational resources required in the quantization process. In fact, as stated in Section 5.05.4.1 and shown in (5.35) the perceptual quantization step ( $\Delta^{perc}$ ) is obtained by multiplying the basic quantization step with the corresponding JND threshold ( $JND_{LM}$  or IDQ in this case). By using the idQP profile representation, the perceptual quantization step is obtained by a simple addition to the basic quantization step (as shown in (5.35)). Provided that additions are less complex than multiplications, what is stored now is only the idQP profile values and then the perceptual QP can be computed on the fly.

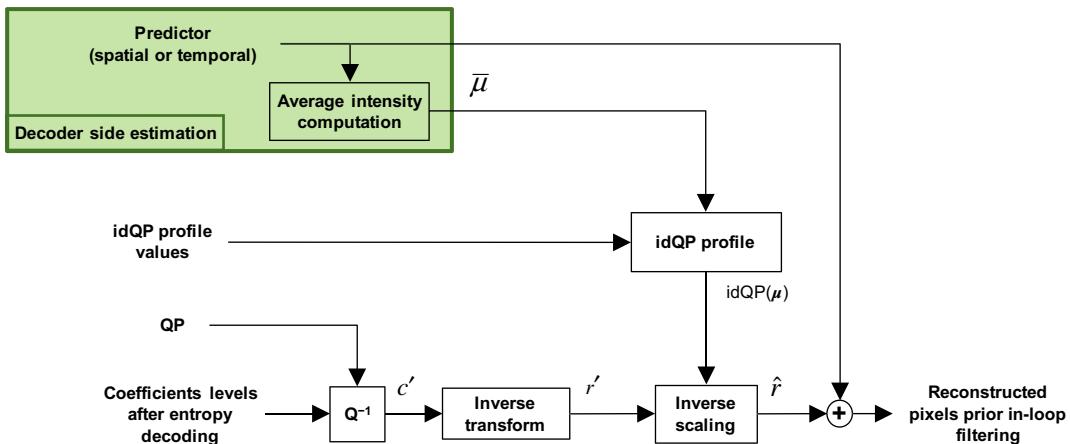
### 5.05.5.3.2 Low computational complexity implementation

For each of the presented IDQ variants, the term needed to be computed at the block level is the average pixel intensity  $\mu$ . As already mentioned, the IDQ tool uses the decoder side intensity computation proposed in [44] where  $\mu$  is computed from the average pixel intensity from the block predictor used for predictive coding (i.e., either intra or inter predictor). The rationale behind the  $\mu$  computation from the predictor, is that first order statistics such as the average can be recovered by first order predictors (either spatial or temporal). The block diagram of the decoder side estimator for  $\mu$  is shown in Figure 5.12. The inverse quantization step is now dependent on the availability of the block predictor.

**FIGURE 5.12**

Decoder block scheme including the decoder side estimation of block intensities.

This dependence may introduce latency in some practical decoder implementations which rely on parallel processing. For example, considering the computation of the motion compensated predictor, in the decoder's pipeline the predictor may become available only before actual reconstruction, i.e., after inverse quantization and inverse transform of the related blocks. In order to avoid this dependency, the perceptual quantization from the transform domain can be moved to the spatial domain using an Intensity Dependent Spatial Quantization (IDSQ) [61]. Since this dependency does not exist at the encoder side, the IDSQ can operate at the decoder side only, as depicted in Figure 5.13. In particular, over the residuals reconstructed after inverse quantization ( $c'$  in the figure), the inverse transform can

**FIGURE 5.13**

Decoder block scheme for the Intensity Dependent Spatial Quantization (IDSQ).

now be applied to obtain the perceptually scaled residual  $r'$ . As may be noted, the inverse quantization for the transform coefficients is independent from the block predictor availability thus no further latency is introduced. The perceptual (inverse) quantization is done in the “*inverse scaling*” that provides the reconstructed residuals  $\hat{r}$  which are then added back to the predictor to reconstruct the coded block. The “*inverse scaling*” module is the same as the inverse uniform quantizer (used in the main inverse quantization step denoted as  $Q^{-1}$  in the figure) but operating in the spatial domain. The reuse of the same inverse quantizer avoids adding new hardware to the codec architecture. At the encoder side, the perceptual quantization is carried out in the same way as per the IDQ tool since, in order to compute the residuals, the predictor must be available anyway.

#### 5.05.5.3 IDSQ performance assessment

Both the IDQ and IDSQ are perceptual coding tools with low-complexity implementation. Moreover, the IDSQ tool extends the IDQ tool to solve the aforementioned latency in the decoding processing. The IDSQ tool is therefore a practical perceptual coding tool which meets requirements for adapting to picture characteristics and maintaining low-complexity implementation. Hence, in this section the rate-distortion performance brought by IDSQ is presented and discussed. The IDSQ tool has been integrated in the HEVC test model (HM, version 7.0) and its performance has been assessed against the HEVC codec. The IDQ profile selected for the tests is the polynomial model from (5.8). In particular, by the rewriting Eq. (5.8) as follows:

$$\text{IDQ}(\mu) = \begin{cases} k_1 \cdot \left(1 - \frac{2\mu}{256}\right)^{\lambda_1} + 1 & \mu \leq 128, \\ k_2 \cdot \left(\frac{2\mu}{256} - 1\right)^{\lambda_2} + 1 & \text{otherwise,} \end{cases} \quad (5.37)$$

it can be noted that the parameters  $k_1$ ,  $k_2$ ,  $\lambda_1$ , and  $\lambda_2$  can be changed to define new profiles. The luminance level  $l$  in (5.8) is now substituted by the pixel intensity  $\mu$ . In particular the pixel intensity can be represented by the luma levels. For the IDSQ assessment, the values for the profile in (5.37) have been made dependent on the quantization parameter used. This dependency is necessary since for high QP values, more gentle IDSQ profiles are needed in order to keep the quality of video coded with this perceptual tool equal to that of video coded with the HEVC codec. The values for the parameters  $k_1$ ,  $k_2$ ,  $\lambda_1$ , and  $\lambda_2$  are listed in Table 5.1.

**Table 5.1** Parameters Dependent on the QP Value Used for IDSQ Profile Computation in Formula (5.37)

Parameter	QP			
	22	27	32	37
$k_1$	4	4	1	0.4
$k_2$	0.8	0.8	0.4	0.2
$\lambda_1$	3	3	3	3
$\lambda_2$	2	2	2	2

The test conditions and videos used in this test follow the common test conditions used in the Joint Collaborative Team on Video Coding (JCT-VC) during the HEVC development. The JCT-VC common test conditions [62] specify four coding configurations: Random Access, All Intra, Low Delay B, and Low Delay P. Each one of these configurations can be run in the Main or High Efficiency 10-bit (HE10) modality. The HE10 modality uses 10-bit processing bit-depth when encoding video content while the Main modality uses only 8-bit processing. Furthermore, in the HE10 coding the Adaptive Loop Filter (ALF), Transform Skip (TS), Non-square Quad Tree (NSQT), and Chroma from Luma model (LM mode) are used as additional coding tools. Moreover, the JCT-VC test material consists of six classes of sequences: A, B, C, D, E, and F. Class A with four sequences at  $2560 \times 1600$  and Class B with five at  $1920 \times 1080$  represent high resolution content oriented towards high-quality broadcasting and video on demand services. Class C with four at  $832 \times 480$  and Class D with four at  $416 \times 240$  represent lower resolution content. Class E with three at  $1280 \times 720$  represents content typical for video conferencing applications. Finally, Class F includes four at different resolutions that contain computer generated objects.

Coding with IDSQ applied to HEVC assumes preservation of visual quality compared with the underlying HEVC configurations in test points. The bitrate reductions provided by the use of IDSQ with respect to the HEVC codec are reported in [Table 5.2](#) where the highest gains for the whole test set are summarized. Only the points with the highest gains for each sequence among a number of test points are reported. As may be noted, IDSQ achieves bitrate reductions of up to 25% and the

**Table 5.2** Bitrate Reductions Between the Sequences Coded with the IDSQ Tool and the Ones Coded with the HEVC Codec in Same Test Points

Sequence name	Configuration	QP	ΔRate [%]	ΔPSNR [dB]
SteamLocomotive	Low Delay B HE10	22	-25.13	-0.56
ChinaSpeed	Low Delay P HE10	22	-19.72	-2.09
Cactus	Low Delay B HE10	22	-16.73	-0.29
Kimono	All Intra HE10	22	-15.39	-0.72
Nebuta	Low Delay B HE10	27	-10.66	-1.03
ParkScene	All Intra HE10	22	-10.05	-0.86
BasketballDrive	Low Delay B HE10	22	-9.57	-0.18
Kimono	Low Delay B Main	27	-9.47	-0.45
Traffic	All Intra Main	22	-8.28	-1.03
ParkScene	Low Delay P Main	22	-8.18	-0.39
KristenAndSara	Low Delay B HE10	22	-7.59	-0.20
PeopleOnStreet	Low Delay B HE10	22	-6.99	-0.37
BQTerrace	Random Access Main	22	-6.79	-0.19
Johnny	Low Delay B HE10	22	-6.58	-0.11

highest reduction is achieved for the *Steam Locomotive* sequence which contains large portions of dark and bright areas. As explained in [Section 5.05.3.2](#), in dark and bright areas the HVS sensitivity to coding artifacts is low and therefore the quantization step is increased to save coding bitrate. Using IDSQ the highest bitrate reductions are obtained for the lower QP values. This is because at high QP values the frequency coefficients are quantized to zero regardless of the additional perceptual quantization leaving little room for compression improvement using IDSQ. For the HEVC codec with the IDSQ tool, the encoder and decoder running times are the same as those for the HEVC codec. This result further supports the claim made at the beginning of this section, i.e., by considering the luminance masking phenomenon only, perceptual video coding can be obtained with low implementation complexity.

For video coding standards such as HEVC, perceptual quantization based on intensity masking can be also achieved with normative functionalities such as the block level quantization step adjustment (see [Section 5.05.4.4](#)). By using this functionality, which is called delta QP (dQP) in HEVC, the quantization step variations, selected on a block basis according to the IDQ or idQP profile at the encoder, are transmitted to the decoder. Therefore, the use of the dQP tools increases the bitrate and, as stated previously, does not allow perceptual adjustment of the quantizer at the transform block level but only at a minimum  $8 \times 8$  block size. In order to quantify the bitrate increment brought by using dQP to perform perceptual quantization instead of using the decoder side estimation of a block's intensity, an additional experiment has also been conducted. The  $QP^{perc}$  selection was performed by using the dQP as specified by the HEVC standard with a minimum block size for QP variation equal to  $8 \times 8$ . Therefore, for each coded block (denoted as a coding unit in HEVC), the average luminance intensity  $\mu$  is computed over the pixels in the original image. The average is then used to select the idQP value using the idQP profile for the luma component only, since the QP for chroma is derived by the one for luma according to a rule specified in the HEVC syntax [18]. The varying QP values are transmitted to the decoder using the dQP normative tool. It should be noted that decoder side estimation is not required since the QP variations are transmitted. For the HEVC codec which uses the dQP tool, the bitrate difference with respect to the codec equipped with the IDSQ tool has been measured and expressed as Bjøntegaard Delta (BD) [63]. The BD is an integral difference between two rate-distortion curves: one for the reference codec and the other for the codec being tested. PSNR was used for this experiment since the two codecs are performing the same kind of perceptual quantization. The integral difference may be expressed with respect to the bitrate (BD-rate), i.e., bitrate reductions for the same PSNR value, or with respect to the PSNR (BD-PSNR), i.e., the PSNR difference at the same rate. Negative values of BD-rate indicate bitrate reductions between the tested and the reference codec results. In the BD-rate measurement, the codec which performs perceptual quantization using the dQP is assumed as the reference, and the tested codec is the HEVC codec with the IDSQ tool added. The BD-rate results for this experiment are listed in [Table 5.3](#). In this table some classes of videos are not included in some configurations. This is because the JCT-VC common test conditions specify only a subset of video classes for some coding configurations. It can be noted that the use of the decoder side estimation allows an average bitrate saving of about 3%. Moreover, the HEVC codec which uses dQP can only vary the quantization parameter at a minimum block size of  $8 \times 8$  while the codec which uses the IDSQ allows a  $4 \times 4$  minimum block size.

**Table 5.3** BD-rates Assuming as Reference the HEVC Codec which Performs Perceptual Quantization Using DQP

	Y (%)	U (%)	V (%)	Y (%)	U (%)	V (%)
	All Intra Main			All Intra HE10		
Class A	-4.3	-1.8	-1.3	-4.3	-1.6	-1.3
Class B	-2.8	-0.8	-0.4	-2.8	-0.9	-0.6
Class C	-2.8	-1.3	-1.3	-2.8	-1.5	-1.4
Class D	-2.4	-0.9	-0.8	-2.3	-0.9	-0.9
Class E	-4.6	-1.4	-1.1	-4.5	-1.5	-0.8
<b>Overall</b>	-3.2	-1.1	-0.9	-3.2	-1.2	-1.0
	Random Access Main			Random Access HE10		
Class A	-3.5	-3.5	-3.2	-3.6	-3.4	-3.0
Class B	-2.8	-3.4	-2.8	-2.8	-3.5	-2.9
Class C	-2.7	-2.6	-2.6	-2.7	-2.8	-2.3
Class D	-2.6	-3.0	-3.2	-2.4	-3.0	-3.1
<b>Overall</b>	-2.8	-3.1	-2.9	-2.8	-3.2	-2.8
	Low delay B Main			Low delay B HE10		
Class B	-3.4	-5.1	-5.1	-3.3	-5.2	-4.1
Class C	-3.3	-4.1	-4.1	-3.1	-4.0	-4.1
Class D	-3.3	-4.6	-5.2	-2.9	-4.6	-4.5
Class E	-3.3	-5.7	-4.2	-3.4	-6.6	-6.3
<b>Overall</b>	-3.3	-4.8	-4.7	-3.2	-5.0	-4.6

## 5.05.6 Conclusions and considerations for future research directions

This chapter has presented a review of enabling technologies and solutions for perceptual video coding targeting block-based motion compensated video codec architectures which perform quantization in the transform domain. Such architectures are widely used and their capacity to deliver better quality or to reduce the required bandwidth, is dependent on compression technology that has been optimized for over two decades. Solutions for further improvement of compression in such codecs lie in perceptual video coding.

Currently, the integration of perceptual coding tools in practical video coding schemes is mainly constituted by the use of a limited number of tools, such as quantization matrices and block level quantization step adjustment. These two tools offer a flexible selection of the model to be used in perceptual coding but do not take into account all the spatial and temporal masking phenomena. As showed in the review of perceptual video codecs, addressing the additional HVS properties can be beneficial in

reducing the bitrate. Additionally, a finer level of adaptation without introducing undesirable side information overhead can be achieved by decoder side estimations. It has been shown that low-complexity solutions already exist, paving a way for next generation codecs with integrated perceptual tools.

In addition to the advances presented in perceptual coding, further investigation of HVS modeling may be beneficial for achieving even higher compression. In fact, the models presented in [Section 5.05.3](#) have been derived from subjective experiments which considered simple visual stimuli such as sinusoidal gratings. In real images and videos there are more complex visual stimuli which cannot be approximated by a combination of the basic gratings such as sinusoidal or Gabor ones. Therefore, several of the models proposed in the literature, namely those for the frequency and contrast masking, should be extended and assessed in more realistic conditions than the ones considered previously. These improved HVS models should also address the *supra-threshold* condition and consider the chroma component of video sequences.

The design of new objective quality assessment metrics, well correlated with the perceived quality, is another area which deserves further studies. The benefit brought by having such objective quality assessment metrics is twofold. From one side, the perceptual quality of coded video may be improved using these metrics in the RDO. From the other side, the evaluation and comparison of perceptual coding tools would become easier as subjective viewings may be substituted by automated procedures which use these improved metrics. This latter aspect is important because the lack of common assessment metrics and test conditions makes the comparison of different perceptual video coding solutions difficult.

In conclusion, there are several interesting extensions to the state-of-the-art in perceptual video coding which can be investigated to achieve further compression and improved quality of coded videos. Such extensions would enable delivery of the large volume of data associated with the increasing spatial and temporal resolutions of future generation video content.

## List of useful websites

Name	Description	Link
Video Quality Expert Group (VQEG)	VQEG performs subjective video quality experiments, validates objective video quality models, and collaboratively develops new techniques	<a href="http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx">http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx</a>
Subjective video quality evaluation	Methodology for the subjective assessment of the quality of television pictures	<a href="http://www.itu.int/rec/R-REC-BT.500/en">http://www.itu.int/rec/R-REC-BT.500/en</a>
MPEG video coding standards	Details of video compression standards from MPEG group	<a href="http://mpeg.chiariglione.org/technologies/media-coding/2-d-video-coding">http://mpeg.chiariglione.org/technologies/media-coding/2-d-video-coding</a>
High Efficiency Video Coding (HEVC)	Useful websites with references to resources related to the HEVC standard	<a href="http://hevc.kw.bbc.co.uk/">http://hevc.kw.bbc.co.uk/</a> , <a href="http://hevc.hhi.fraunhofer.de/">http://hevc.hhi.fraunhofer.de/</a>

## Glossary

<b>Discrete cosine transform</b>	the Discrete Cosine Transform (DCT) is a transformation which operates over signals with finite number of samples. The transformation expresses each sample as the summation of cosine functions oscillating at different frequencies. The DCT is widely used in image and video compression given its property of energy compaction, i.e., the representation of one signal with a few number of significant coefficients. The DCT belongs to the class of Fourier-related transforms and uses only real numbers to compute the coefficients
<b>Entropy coding</b>	a lossless process used in data compression to represent a given set of data with a fewer number of bits with respect to its original coding. The main idea behind each entropy coding scheme is the replacement of bits, symbols of data series occurring in the source data with different codes that are statistically derived. New codes can comprise a number of bits (variable length codes) or fractional bits (as in arithmetic coding). The length of each code is inversely proportional to the probability of occurrence of the associated part of the source data. For example, less frequent symbols will receive a longer codeword and vice versa
<b>Motion compensation</b>	a technique commonly used in video codecs belonging to the MPEG-x and H.26x families. In motion compensation, selected pixels of a video frame are described in terms of the displacement with respect to a reference frame. The reference frame can be (temporally) located in the past or in the future with respect to the frame where motion compensation is performed. The rationale behind motion compensation is that in many real video sequences the differences between one frame and the following one are only due to camera or object movements. To improve compression efficiency only these differences can be sent to the decoder, leaving all the pixel values belonging to nonmoving image areas the same as in previous frames
<b>Quantization</b>	the process of mapping a large set of values into a smaller set of values with lower precision. Quantization is performed by a device or algorithm called quantizer. The precision point at which the input values are represented is called quantization step. Small quantization step values lead to a finer representation of the input values, while coarse quantization step values map several input values to a unique output value (also called reconstruction value). The error introduced by a quantizer is usually called quantization or rounding error
<b>Video coding standard</b>	the format which specifies the set of rules to decompress a video bitstream and obtain the final decoded video. The set of rules is specified in terms of syntax and semantics of each bit or set of bits read from the bitstream. The video coding standard describes only the decoder operations while leaving unspecified the encoder although it is implicitly assumed that this latter produces a bitstream compliant with the syntax and semantics specified in the standard
<b>Human visual system</b>	the Human Visual System (HVS) is a simplified model of the biological and psychological processes which take part in the human eye and brain to form and process the information related to an observed scene

<b>Contrast sensitivity function</b>	the Contrast Sensitivity Function (CSF) expresses the sensitivity of the human visual system to changes in luminance or color in the observed visual stimuli. The sensitivity can be measured in different domain as for example the commonly used frequency domain which allows to quantify the sensitivity to changes in contrast at different frequency values
--------------------------------------	---

## References

- [1] ITU-R, Parameter Values for Ultra-High Definition Television Systems for Production and International Programme Exchange, Rec. BT.2020, August 2012.
- [2] NHK Science and Technology Research Laboratories, Super high vision format. <<http://www.nhk.or.jp/strl/english/aboutstrl1/r1-1-1.htm>> (retrieved October 2012).
- [3] British Broadcasting Corporation–Research and Development (BBC R&D) Blog. <<http://www.bbc.co.uk/blogs/researchanddevelopment/2012/08/the-olympics-in-super-hi-visio.shtml>> (retrieved on October 2012).
- [4] ITU-T Rec. H262 & ISO/IEC 13818–2 MPEG-2, Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video, November 1994.
- [5] ITU-T Rec. H.264 & ISO/IEC 14496–10 (MPEG-4 AVC), Advanced Video Coding for Generic Audiovisual Services, Version 16, January 2012.
- [6] K. McCann, Progress Towards High Efficiency Video Coding DVB Scene, No. 39. <[http://issuu.com/dvbscene/docs/dvb\\_scene\\_issue\\_39](http://issuu.com/dvbscene/docs/dvb_scene_issue_39)> (retrieved October 2012), March 2012.
- [7] T. Wiegand et al., Special section of the joint call for proposals on high efficiency video coding (HEVC) standardization, IEEE Trans. Circuits Syst. Video Technol. 20 (12) (2010) 1661–1666.
- [8] H.R. Wu, K.R. Rao, Digital video image quality and perceptual coding, CRC Press, November 2005, p. 640.
- [9] N. Jayant, J. Johnston, R. Safranek, Signal compression based on models of human perception, Proc. IEEE 81 (10) (1993) 1385–1422.
- [10] G.J. Sullivan, T. Wiegand, Rate-distortion optimization for video compression, IEEE Signal Process. Mag. 15 (6) (1998) 74–90.
- [11] Z. Wang, A.C. Bovik, Mean square error: love it or leave it? A new look at signal fidelity measures, IEEE Signal Process. Mag. 26 (1) (2009) 98–117.
- [12] B. Girod, What's wrong with mean squared error? in: What's wrong with mean squared error? A.B. Watson (Ed.), Visual Factors of Electronic Image Communications, MIT Press, 1993, pp. 207–220.
- [13] S. Winkler, P. Mohandas, The evolution of video quality measurement: from PSNR to hybrid metrics, IEEE Trans. Broadcast. 54 (3) (2008) 660–668.
- [14] K. Seshadrinathan, A.C. Bovik, Automatic prediction of perceptual quality of multimedia signals – a survey, Springer Int. J. Multimedia Tools Appl. 51 (1) (2011) 163–186.
- [15] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
- [16] Y.-H. Huang, T.-S. Ou, P.-Y. Su, H.H. Chen, Perceptual rate-distortion optimization using structural similarity index as quality metric, IEEE Trans. Circuits Syst. Video Technol. 20 (11) (2010) 1614–1624.
- [17] P. List, A. Joch, J. Lainema, G. Bjørtegaard, M. Karczewicz, Adaptive deblocking filter, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 614–619.
- [18] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, High efficiency video coding (HEVC) text specification draft 8, JCTVC-J1003, in: 10th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Stockholm, SE, July 2012.

- [19] C.-H. Chou, Y.-C. Li, A perceptually tuned subband image coder based on the measure of just-noticeable distortion profile, *IEEE Trans. Circuits Syst. Video Technol.* 5 (6) (1995) 467–476.
- [20] X.K. Yang, W.S. Ling, Z.K. Lu, E.P. Ong, S.S. Yao, Just noticeable distortion model and its applications in video coding, *Signal Process. Image Commun.* 20 (7) (2005) 662–680.
- [21] J.L. Mannos, D.J. Sakrison, The effect of a visual fidelity criterion in the encoding of images, *IEEE Trans. Inf. Theory* 2 (4) (1974) 525–536.
- [22] K.N. Ngan, K.S. Leong, H. Singh, Adaptive cosine transform coding of image in perceptual domain, *IEEE Trans. Acoust. Speech Signal Process.* 37 (11) (1989) 1743–1750.
- [23] Z. Wei, K.N. Ngan, Spatio-temporal just noticeable distortion profile from grey scale image/video in DCT domain, *IEEE Trans. Circuits Syst. Video Technol.* 19 (3) (2009) 337–346.
- [24] A.J. Ahumada, H.A. Peterson, Luminance-model-based DCT quantization for color image compression, in: *Proceedings of SPIE: Human Vision, Visual Processing and Digital Display III*, vol. 1666, 1992.
- [25] A.B. Watson, G.Y. Yang, A. Solomon, J. Villasenor, Visibility of wavelet quantization noise, *IEEE Trans. Image Process.* 6 (8) (1997) 1164–1175.
- [26] X. Zhang, W. Lin, P. Xue, Improved estimation for just-noticeable visual distortion, *Signal Process.* 85 (4) (2005) 795–808.
- [27] J.M. Foley, G.M. Boynton, A new model of human luminance pattern vision mechanism: analysis of the effects of pattern orientation, spatial phase and temporal frequency, in: *Proceedings of SPIE: Computational Vision Based on Neurobiology*, vol. 2054, 1994.
- [28] I. Höntsche, L.J. Karam, Adaptive image coding with perceptual distortion control, *IEEE Trans. Image Process.* 11 (3) (2002) 312–322.
- [29] R. Leung, D. Taubman, Perceptual optimization for scalable video compression based on visual masking principles, *IEEE Trans. Circuits Syst. Video Technol.* 19 (3) (2009) 337–346.
- [30] M. Naccari, F. Pereira, Comparing spatial masking modeling in just noticeable distortion controlled H.264/AVC video coding, in: *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services*, Desenzano del Garda, IT, April 2010.
- [31] S. Daly, *Engineering observations from spatio-velocity and spatio-temporal visual models*, in: *Vision Models and Applications to Image and Video Processing*, Kluwer Academic Press, 2001, p. 229.
- [32] J.G. Robson, Spatial and temporal contrast-sensitivity functions of the visual system, *J. Opt. Soc. Am.* 56 (10) (1966) 1141–1142.
- [33] D.H. Kelly, Motion and vision II: stabilized spatio-temporal threshold surface, *J. Opt. Soc. Am.* 69 (10) (1979) 1340–1349.
- [34] E.Y. Lam, J.W. Goodman, A mathematical analysis of the DCT coefficient distribution for images, *IEEE Trans. Image Process.* 9 (10) (2000) 1661–1666.
- [35] M.G. Ramos, S. Hemami, Suprathreshold wavelet coefficient quantization in complex stimuli: psychophysical evaluation and analysis, *J. Opt. Soc. Am.* 18 (10) (2001) 1–13.
- [36] M. Naccari, C. Brites, J. Ascenso, F. Pereira, Low complexity deblocking filter perceptual optimization for the HEVC codec, in: *Proceedings of the IEEE International Conference on Image Processing*, Brussels, BE, September 2011.
- [37] Test Model 5, ISO/IEC, JTC/SC29/WG11/N0400, 1993.
- [38] K. Sato, On LBS and quantization, JCTVC-D308, in: 4th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Daegu, KR, January 2011.
- [39] C.-H. Chou, C.-W. Chen, A perceptually optimized 3-D subband image codec for video communication over wireless channels, *IEEE Trans. Circuits Syst. Video Technol.* 6 (2) (1996) 143–156.
- [40] C.-M. Mak, K.N. Ngan, Enhancing compression rate by just noticeable distortion model for H.264/AVC, in: *Proceedings of the IEEE International Symposium on Circuits and Systems*, Taipei, TW, May 2009.

- [41] Y. Jia, W. Lin, A.A. Kassim, Estimating the just-noticeable distortion for video, *IEEE Trans. Circuits Syst. Video Technol.* 16 (7) (2006) 820–829.
- [42] Z. Chen, C. Guillemot, Perceptually-friendly H.264/AVC video coding based on foveated just-noticeable-distortion model, *IEEE Trans. Circuits Syst. Video Technol.* 20 (6) (2010) 806–819.
- [43] Z. Wang, L. Lu, A.C. Bovik, Foveation scalable video coding with automatic fixation selection, *IEEE Trans. Image Process.* 12 (2) (2003) 243–254.
- [44] M. Naccari, F. Pereira, Advanced H.264/AVC-based perceptual video coding: architecture, tools, and assessment, *IEEE Trans. Circuits Syst. Video Technol.* 21 (6) (2011) 766–782.
- [45] M. Naccari, F. Pereira, Integrating a spatial just noticeable distortion model in the under development HEVC codec, in: Proceedings of the IEEE International Conference on Acoustic Speech and Signal Processing, Prague, CZ, May 2011.
- [46] P. Ndjiki-Nya, D. Doshkov, H. Kaprykowsky, F. Zhang, D. Bull, T. Wiegand, Perception-oriented video coding based on image analysis and completion: a review, *Signal Process. Image Commun.* 27 (6) (2012) 579–594.
- [47] A. Khandelia, S. Goreche, B. Lall, S. Chadhury, M. Mathur, Parametric video compression scheme using AR based texture synthesis, in: Proceedings of the International Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, IN, December 2008.
- [48] A. Stajanovic, M. Wien, J.-R. Ohm, Dynamic texture synthesis for H.264/AVC inter coding, in: Proceedings of the IEEE International Conference on Image Processing, San Diego, CA, USA, October 2008.
- [49] A. Stajanovic, M. Wien, T.K. Tan, Synthesis-in-the-loop for video texture coding, in: Proceedings of the IEEE International Conference on Image Processing, Cairo, EG, October 2009.
- [50] D. Doshkov, P. Ndjiki-Nya, H. Lakshman, M. Köppel, T. Wiegand, Towards efficient intra prediction based on image inpainting methods, in: Proceedings of the Picture Coding Symposium, Nagoya, JP, December 2010.
- [51] A.A. Efros, T.K. Leung, Texture synthesis by nonparametric sampling, in: Proceedings of the IEEE International Conference on Computer Vision, Corfu, GR, September 1999.
- [52] A. Buades, B. Coll, J.-M. Morel, A non-local algorithm for image denoising, in: Proceedings of the IEEE Computer Vision and Pattern Recognition, San Diego, CA, USA, June 2005.
- [53] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, *IEEE Trans. Image Process.* 13 (9) (2004) 1200–1212.
- [54] T.K. Tan, C.S. Boon, Y. Suzuki, Intra prediction by average template matching, in: Proceedings of the IEEE Consumer Communications and Networking Conference, Las Vegas, NE, USA, January 2007.
- [55] J.Y.A. Wang, E.H. Adelson, Representing moving image with layers, *IEEE Trans. Image Process.* 3 (5) (1994) 625–638.
- [56] A. Dumitraş, B.G. Haskell, An encoder-decoder texture replacement method with application to content-based movie coding, *IEEE Trans. Circuits Syst. Video Technol.* 14 (6) (2004) 825–840.
- [57] F. Zhang, D.R. Bull, A parametric framework for video compression using region-based texture models, *IEEE J. Sel. Top. Sign. Process.* 5 (7) (2011) 1378–1392.
- [58] P. Ndjiki-Nya, T. Hinz, T. Wiegand, Generic and robust video coding with texture analysis and synthesis, in: Proceedings of the IEEE International Conference on Multimedia and Expo, Beijing, CN, July 2007.
- [59] M. Naccari, M. Mrak, On just noticeable distortion quantization in the HEVC codec, JCTVC-H0477, in: 8th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), San José, CA, USA, February 2012.
- [60] M. Naccari, M. Mrak, D. Flynn, A. Gabriellini, On just noticeable distortion quantization in the HEVC codec, JCTVC-I0257, in: 9th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Geneva, CH, April-May 2012.
- [61] M. Naccari, M. Mrak, D. Flynn, A. Gabriellini, Improving HEVC compression efficiency by intensity dependent spatial quantization, JCTVC-J0076, in: 10th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Stockholm, SE, July 2012.

- [62] F. Bossen, Common HM test conditions and software reference configurations, JCTVC-H1100, in: 8th Meeting of the Joint Collaborative Team on Video Coding (JCT-VC), San José, CA, USA, February 2012.
- [63] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, VCEG-M33, in: 33rd Meeting of the Video Coding Expert Group (VCEG), Austin, TX, USA, April 2001.

# How to Use Texture Analysis and Synthesis Methods for Video Compression

# 6

**Dimitar Doshkov and Patrick Ndjiki-Nya**

*Image Processing Department, Fraunhofer Institute for Telecommunications – Heinrich-Hertz-Institute, 37 Einsteinufer, Berlin D-10587, Germany*

## Nomenclature

### Mathematical notations

<i>scalar</i>	small and italic
<i>vector</i>	small, italic, and bold
<i>Matrix</i>	capital, italic, and bold
<i>function(·)</i>	small, italic, and brackets
<i>Set</i>	capital and script

## Symbols

### Symbols

$I$

### Denotation

samples in the reconstructed image plane

$\hat{I}$

unknown (to-be-predicted) samples over  $\Omega$

$\Omega$

current (to-be-coded) block

$\partial\Omega$

spatial neighbors (one sample) of the current block

$a_{i,j}$

prediction coefficients

$\mathbf{a}$

vector comprising the prediction coefficients  $\{a_{i,j}\}$

$c_h, c_v$

horizontal and vertical orders of the AR/LS-based model

$c$

number of prediction coefficients (model order)

$s$

number of samples in the training area

$\mathbf{X}$

neighboring sample matrix for each of the samples in  $\mathbf{y}$

$\mathbf{y}$

samples in the training area (vectorized)

$\varepsilon(\cdot)$

innovation signal

$\sigma^2$

variance of  $\varepsilon$

$n(\cdot)$

white Gaussian noise

$err$

prediction error of the AR model

$P$

input of the ARMA model

$S$

video sequence

$S_m$

mean of all images used for the training of the ARMA model

$A, B, D$	parametric matrices of the ARMA model
$V, W$	independent and identically distributed random variables
$\nabla$	nabla operator (gradient operator)
$\Delta$	Laplace operator
$\lambda_e$	Lagrange multiplier
$\mathcal{P}, \mathcal{P}^*$	2D texture pattern (templates)
$\mathcal{N}$	neighborhood system
$g(\cdot)$	cost function of graph-cut synthesis algorithm
$\mathcal{A}, \mathcal{B}$	2D + $t$ texture patterns (templates)
$x, y, t, l, k$	sample positions in an image/video
$q_s(\cdot), q_t(\cdot)$	spatial and temporal video quality
$f_\beta$	linear, anisotropic gradient filter (e.g. Sobel) of orientation $\beta$

### 5.06.1 Introduction

All areas of multimedia communication and storage have significantly benefited from advances in video coding technology. Some impressive application examples are mobile video, mobile TV, DTV, 3DTV, high definition (HD) TV, DVD/Blue-ray players, digital cameras, video telephony, video conferencing, Internet video/streaming, and multimedia messaging. None of these would operate effectively without high performance compression algorithms.

Since the early 1990s, when the technology was in its infancy, international video coding standards such as H.261, MPEG-1, MPEG-2/H.262, H.263, and MPEG-4 Part 2 have been powerful engines behind the commercial success of digital video compression. Today, the standard H.264/AVC [1] is the universally deployed format in video coding. Its coding efficiency is reportedly well above any other video coding standard [2]. The question arises, how coding efficiency could be improved beyond the efficiency that can be achieved with the current H.264/AVC design. One such approach is the most recently standardized High Efficiency Video Coding (HEVC) project [3]. In HEVC, significant coding gains are achieved mainly by a generalization of the methods in H.264/AVC. HEVC mainly focuses on two key issues [3]: to address increased resolution ranges between QVGA ( $320 \times 240$ ) and Ultra HDTV ( $7680 \times 4320$ ), and to use parallel processing design. Coding strategies, as described in this chapter, can be applied on top of existing hybrid video coding designs—including HEVC.

In this work, we will analyze and describe the techniques behind the most promising perception-oriented coding strategies based on texture analysis and synthesis. An overview on gains to be expected from such methods will be provided additionally.

The remainder of the paper is organized as follows: A review of different coding strategies based on texture analysis and synthesis is given in [Section 5.06.2](#). In [Section 5.06.3](#) leading block-based video coding techniques are detailed. Furthermore, the integration of the described approaches into an H.264/AVC and an HEVC video codec is addressed. A detailed presentation of a region-based video coding framework that is used as a basis structure for explanations is given in [Section 5.06.4](#). Here, the modules required in the proposed framework are introduced, their necessity is illustrated, and their interactions are explained. Alternative state-of-the-art approaches are presented with respect to that baseline. Finally, concluding thoughts are given in [Section 5.06.5](#).

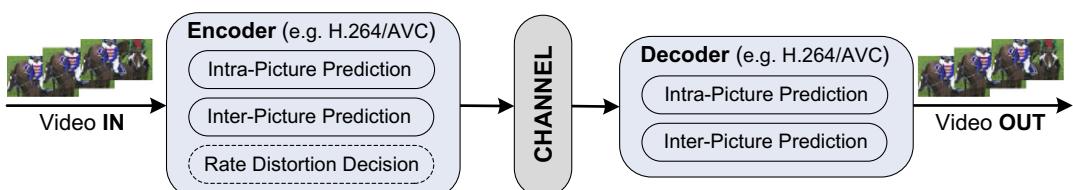
## 5.06.2 Perception-oriented video coding strategies

Perception-oriented video coding strategies based on texture analysis and synthesis (TAS) can be divided in two main categories [4]: (1) methods that are fully compatible with hybrid block-based codecs and (2) alternative codec designs such as region-based coding.

Block-based coding—the fully compatible approach—corresponds to the integration of TAS-based techniques in hybrid block-based codecs as H.264/AVC or HEVC while targeting a pixel-accurate reconstruction of the input signal. In general, such coding frameworks replace, improve, or extend existing intra and inter prediction modes [1,3] through advanced perception-oriented methods (cf. Figure 6.1). Block-based compression methods perform coding by splitting the current image into blocks a priori. Each block is separately encoded, transmitted, and decoded. Early works, which described new and efficient perceptual-based video coding methods, were presented in [5–26]. An extensive overview of several publications can be found in [4].

“Alternative codec designs” modify the hybrid codec structure to fit the perception-oriented techniques. That is, a globally instead of an MSE accurate texture representation is aimed at. For that, these approaches usually do not transmit residual information. Hence, TAS codecs take advantage of reduced side information compared to an MSE-based representation. As a result, the reconstructed video, although perceptually similar, is slightly different from the original. In general, the region-based coding approaches consist of a texture analyzer at the encoder and a texture synthesizer at the decoder. The texture analyzer thereby identifies the target textures. For these, it is assumed that the viewer perceives the semantic meaning of the displayed texture rather than the specific details therein. For identified textures, the synthesizer regenerates an approximate version of the original based on corresponding meta data transmitted by the encoder. Over the past two decades, many promising region-based video coding methods have been developed [27–36] and thoroughly discussed in [4].

Research has shown that the Mean Squared Error (MSE) criterion, typically used in hybrid video codecs such as H.264/AVC and HEVC, is not an adequate coding distortion measure for high frequency regions when displayed with limited spatial resolution, as these typically yield high coding costs [4]. Therefore the incorporation of new perceptual-based distortion measures instead of MSE is an important aspect of TAS coding, especially for the region-based codecs. In fact, perceptual distortion assessment is needed to find the optimal coded representation of a region in the perceptual rate-distortion sense.



**FIGURE 6.1**

Typical hybrid block-based video codec with integrated perceptual-based TAS tools. The rate-distortion module is marked with a dashed line to emphasize limited technical contributions in this area to-date.

### 5.06.3 Block-based video coding techniques

In order to provide a comprehensive description of the TAS techniques that are used in conjunction with block-based video coding, the main modules of the host codec, where the perceptual-based approaches are integrated, will be highlighted. In the following subsections, different block-based video coding methods will be examined. Their integration into the host codec and their respective performance will be presented.

#### 5.06.3.1 The video coding framework

The hybrid block-based video coding frameworks addressed in this chapter correspond to the video compression standards H.264/AVC and HEVC. Figure 6.1 depicts a simplified block diagram of a hybrid video codec with highlighted modules, where perceptual-based approaches can be integrated. It should be noted that this approach may still yield a standard-conforming bit stream. The rate-distortion module is marked with a dashed line to emphasize that coding performance improvement through optimization of this module via perceptual-based distortion measures is highly limited to date (cf. [16]). Note that this is one of the topics to be investigated in the PROVISION Initial Training Network (ITN) (Marie Curie Program sponsored by the European Union) that will be running from 2013 to 2017. Therefore explanations will be constrained to techniques for intra and inter block prediction in the following discussion.

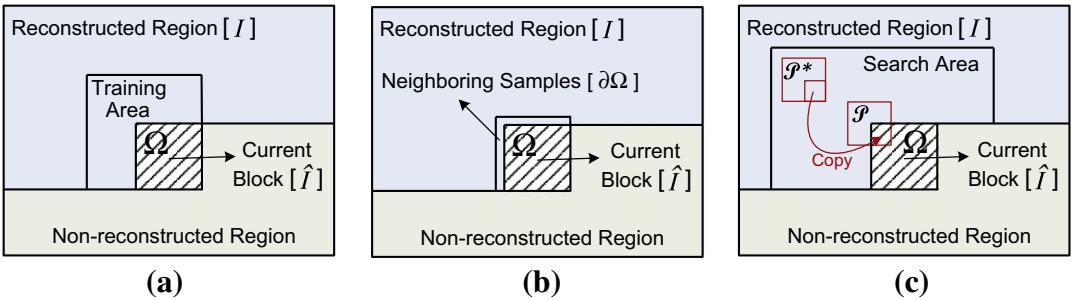
An encoding processing chain would typically be as follows [1,3]:

- (1) Each input picture is split into blocks—a priori, i.e. content-independent segmentation.
- (2) Each block is then separately intra or inter predicted based on rate-distortion (RD) decisions.
- (3) The residual signal of the prediction procedure, which is the difference between the original and its prediction, is transformed.
- (4) The resulting transform coefficients are then scaled, quantized, entropy coded, and transmitted with the prediction information.

At the receiver side, the decoder repeats the same processing steps in an inverse order to that at the encoder, ensuring that the prediction (intra or inter) and final signals are identical.

For better comprehension, we will shortly describe the functionality of the intra and inter prediction modules (cf. [1,3] for more information). Intra prediction is a key tool to eliminate spatial redundancy in video data. The prediction of the current block is formed by using the neighboring, already reconstructed samples (cf. Figure 6.2), i.e. no dependencies on other pictures are available. Intra prediction is performed for different block sizes depending on the used codec standard. In general, intra prediction supports several directional, direct current (DC) and plane prediction modes. The selected prediction mode is transmitted as side information to the decoder, such that the same prediction signal can be generated during the reconstruction procedure. Inter-picture predictive coding exploits the temporal dependency between frames. The encoding process for inter prediction consists of finding appropriate motion data comprising the pre-defined reference pictures and a motion vector (MV) to be applied for predicting the samples of the current block. The decoder generates an identical inter prediction signal by applying motion compensation (MC) using the transmitted MV and mode decision information.

In general, the advanced perceptual-based intra/inter predictors work well as additional modes or as replacements for the existing ones. The coding gains obtained through the new approaches are typically



## **FIGURE 6.2**

Notation conventions for intra prediction with (a) parametric, (b) PDE-based, and (c) non-parametric (template-based) approaches.

achieved either because of better prediction performance and/or because of a reduction in the signalling (side) information.

### 5.06.3.2 Perception-oriented coding techniques

Image completion techniques represent the main components of TAS block-based coding. The generic term “image completion” was specified in the work of Ndjiki-Nya et al. [4] and refers to the seamless reconstruction of missing image parts. Hence, “image completion” is a generalized definition of all approaches known as inpainting, texture synthesis, or image/video restoration. In the literature, image completion techniques can be separated into three different categories: (1) parametric, (2) partial differential equations-based, and (3) non-parametric.

In the last decade, all three image completion categories have been well studied in the context of inter and intra prediction, as potential video compression methods for hybrid block-based video coding. Comparing the results of the different methods is difficult, as they are evaluated using different codec versions and parameter settings [4]. Furthermore, the signalling approaches for the side information and the data are different for each test framework. The definition of an open test and evaluation framework will be addressed in the upcoming PROVISION ITN. Some of the main implementation strategies will nonetheless be explained in detail in the following.

### 5.06.3.3 Parametric approaches

The most reliable and successful parametric techniques that have been used to improve the coding performance of H.264/AVC and recently HEVC are based on the well-known (AR) [5,11] and Auto Regressive Moving Average (ARMA) [6, 7] models. Furthermore, the Least Squares-based (LS) linear prediction model has also been well investigated for inter [8] and intra [9–12] prediction.

The implementation of parametric prediction modes, e.g. in the context of intra prediction, is illustrated in Figure 6.2a.  $I$  corresponds to the samples in the reconstructed image region and  $\hat{I}$  represents the unknown samples in the current block  $\Omega$ . The inter prediction case can be interpreted correspondingly, with the difference that spatio-temporal training data is used. It can be seen that parametric approaches

need a well-defined training area to be properly executed. In order to clarify, why they can be applied to coding, we will explain the Least Square (LS) based linear prediction, the AR and the ARMA models into some more detail in the following.

#### 5.06.3.3.1 LS-based linear prediction

LS-based linear prediction is a mathematical operation where future values of a discrete spatial/temporal signal are estimated as a linear function of previous samples. Assuming that we consider an intra prediction application, this approach is suitable for embedding the local texture characteristics of neighboring samples. Thus the prediction process can be adaptively adjusted to the local context (cf. Figure 6.2a). The most common representation of this method within a spatial (image) plane can be expressed as:

$$\hat{I}(x, y) = \sum_{i=0}^{c_h} \sum_{j=0}^{c_v} a_{i,j} I(x - i, y - j) \quad \text{with } (i, j) \neq (0, 0), \quad (6.1)$$

where  $(x, y)$  represents the position in the current image  $I$ ,  $a_{i,j}$  represent the prediction coefficients, and  $c_h, c_v$ , are the horizontal and vertical orders of the LS-based model respectively (cf. Figure 6.4).

Before (6.1) can be applied to predict samples in the to-be-coded block  $\Omega$  (cf. Figure 6.2a), the prediction coefficients  $a_{i,j}$  have to be determined. Thus, an appropriate training area with known samples is required. Following Markov Random Field (MRF) theory, the most likely texture with similar probability density function is adjacent to the unknown area  $\Omega$ . Figure 6.2a shows an example of a training area that is adjacent to  $\Omega$ .

The optimal prediction coefficients can be estimated as the solution to the following least squares problem:

$$\boldsymbol{a}_{c \times 1} = \underset{\boldsymbol{a}}{\operatorname{argmin}} \| \mathbf{y}_{s \times 1} - \mathbf{X}_{s \times c} \boldsymbol{a}_{c \times 1} \|^2, \quad (6.2)$$

where  $\boldsymbol{a}$  is a vector comprising the prediction coefficients  $\{a_{i,j}\}$ ,  $\mathbf{y}$  denotes the vectorized samples in the training area (cf. Figure 6.2a) and  $\mathbf{X}$  represents the neighboring sample matrix for each of the samples in  $\mathbf{y}$ . Furthermore, the subscripts in (6.2) represent the dimension of the vectors and matrices, where  $c(c = c_h c_v + c_h + c_v)$  is the number of prediction coefficients (cf. Figure 6.4) and  $s$  denotes the number of samples in the training area (the number of linear equations). Equation (6.2) can be estimated by the closed-form solution:

$$\boldsymbol{a} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{y}). \quad (6.3)$$

Note that, the implementation of this approach as an inter predictor can be implemented in an analog way. In this case, a spatio-temporal training area (considering also previously reconstructed frames) has to be utilized.

#### 5.06.3.3.2 AR-based prediction

The Auto Regressive (AR) model is a type of random process which is often used to model and predict various types of natural phenomena. The Auto Regressive model is one of a group of linear prediction formulae that attempt to predict a system output, based on the known samples.

In AR modeling every sample is estimated as a linear combination of neighboring samples in space (and in time) plus a noise term. Hence, it can be considered as an extension of LS-based linear prediction.

The noise term ensures the innovation in the prediction process and is typically represented by a white Gaussian noise ( $n(0, \sigma^2)$ ). Hence, a linear Auto Regressive model (e.g. in the context of intra prediction) can, similarly to (6.1), be expressed as:

$$\hat{I}(x, y) = \sum_{i=0}^{c_h} \sum_{j=0}^{c_v} a_{i,j} I(x - i, y - j) + \varepsilon(x, y) \text{ with } \varepsilon(x, y) \sim n(0, \sigma^2) \text{ and } (i, j) \neq (0, 0), \quad (6.4)$$

where  $\varepsilon$  denotes the innovation signal which is driving the AR model and  $\sigma^2$  represents the variance of  $\varepsilon$ .  $\sigma^2$  can be estimated as described in [11,37]:

$$\sigma^2 = \frac{err}{s} \quad (6.5)$$

with

$$err = \|y_{s \times 1} - X_{s \times c} a_{c \times 1}\|^2. \quad (6.6)$$

The notation used in (6.5) and (6.6) is the same as in the previous section.

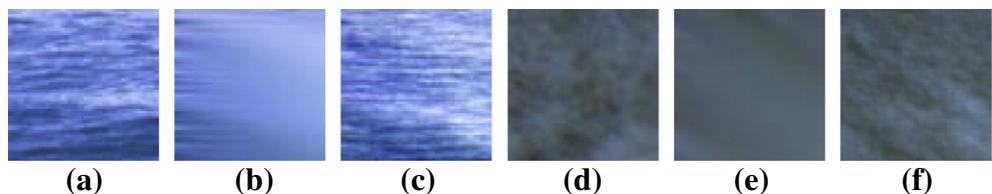
The main difference between LS-based linear prediction and AR modeling lies in their functionality, i.e. in their final results. Figure 6.3 shows a comparison between the visual outcomes generated with these methods. It is apparent that utilizing the AR approach leads to a subjective improvement of the final texture quality. In general, the LS-based linear prediction tends to produce smoother results, especially when the model order ( $c$ ) is relatively low. A thorough tutorial on the AR image completion approach can be found in [37].

#### 5.06.3.3.3 ARMA-based prediction

The ARMA model is a statistical process, in which both autoregressive analysis and moving average methods are applied primarily to time series data. In the video coding scenario, the synthesized textures are generated using the well-established approach developed by Doretto et al. [38]:

$$\begin{cases} P(t+1) = AP(t) + BV(t), \\ S(t) = DP(t) + S_m + W(t), \end{cases} \quad t = 1, 2, \dots, n. \quad (6.7)$$

Here  $P$  is the input to the AR part of the ARMA model and  $S$  is a video sequence with  $n$  images.  $A$ ,  $B$ , and  $D$  are parametric matrices of the model, and  $V$  and  $W$  are assumed as independent and



**FIGURE 6.3**

Visual comparison. (a and d) Original texture of size  $64 \times 64$ , result by (b and e) LS-based linear prediction, and (c and f) AR image completion approach.

identically-distributed random variables.  $S_m$  is the mean of all the training images. Hence, this approach [38] requires a spatio-temporal training area.

The method, introduced by Doretto et al. [38], was initially designed for dynamic texture synthesis and the generation of endless sequences from relatively short video samples. Therefore, to make the ARMA model applicable for video coding purposes, Stojanovic et al. [6] modify the generic approach so that one single prediction picture is generated instead of a random video sequence. Due to the fact that the picture buffer in the encoder and respectively in the decoder is limited, the new synthesized frame bases only on five previously reconstructed frames (30 frames were also tested) as training data. This inter prediction algorithm replaces the oldest frame in the reference picture buffer by the synthetically extrapolated frame using the H.264/AVC framework. Both the encoder and the decoder then use the predicted frame as a reference frame. Theoretically, the ARMA model can also be adapted to intra prediction scenarios. Zhang and Bull [36] further modified this based on pre-warping the frames to be synthesized with improved results.

#### 5.06.3.3.4 Optimization strategies and performance

The integration of the proposed LS/AR-based parametric image completion methods as intra or inter prediction modes can be further optimized. The works of Liu et al. [9], Chen et al. [10], and Doshkov et al. [11] have shown that applying different block-based scanning order predictors results in better intra coding than using only one orientation. The implementation of these prediction modes, in the context of intra prediction, is illustrated in Figure 6.4. It can be seen that three different settings are proposed [11]: general, horizontal, and vertical prediction. It is assumed that the general prediction is beneficial for blocks with predominantly textured content. Likewise, the horizontal prediction is beneficial for blocks containing horizontally oriented edges. The training area and the model structure are designed such that appropriate edge statistics are propagated from the left image side (cf. [11]). The vertical prediction mode is defined to predict dominant vertically oriented edges. The following coding gains were reported for different coding frameworks: for LS-based intra prediction up to 18% [9], up to 7% [10], and more than 7% [11]; for LS-based inter prediction up to 11% [8]; for AR-based intra prediction more than 2% [11].

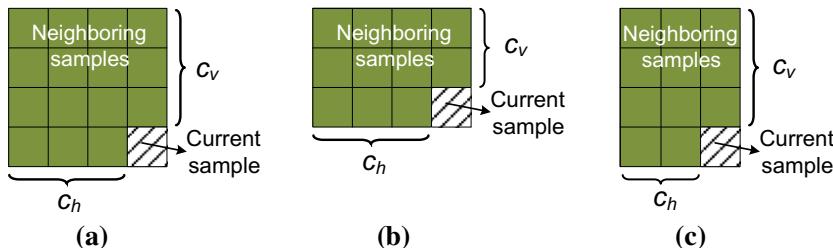


FIGURE 6.4

LS/AR-based models used for (a) general ( $c_h = c_v$ ), (b) horizontal ( $c_h > c_v$ ), and (c) vertical prediction ( $c_h < c_v$ ).

An important work that can be considered as a special case of the block-based video coding techniques with TAS approaches, was developed by Khandelia et al. [5]. Their framework incorporates a spatio-temporal AR model to ensure both spatial as well as temporal completion. The AR technique is used to reconstruct only textured (non-edged) blocks in H.264/AVC. Bit rate savings up to 54% were reported for the “Bridge far” sequence. Note that this framework was subjectively evaluated.

In order to improve the coding efficiency of H.264/AVC, Stojanovic et al. [6] and Chen et al. [7] have further optimized the ARMA implementation proposed in the first framework of Stojanovic et al. (cf. Section 5.06.3.3.3). The newer method by Stojanovic et al. [6] performs multi-pass encoding so that the position of the reference pictures changes during each pass. Hence, the position of the synthesized picture in the decoded picture buffer is adaptively estimated. Bit rate savings up to 15% (“Bridge far”) were obtained in comparison to the genuine H.264/AVC codec. Furthermore, Chen et al. [7] proposed a further optimization of the ARMA model. Their methods can properly handle illumination changes between subsequent frames. The highest bit rate savings that can be achieved with their approach, are higher than 7% for the “Mobisode2” sequence.

### 5.06.3.4 PDE-based approaches

The most promising partial differential equation (PDE) based completion algorithms for compression rely on the well-known Laplace PDE and the total variation (TV) model. Successful adaptations of Laplace PDE to image and video coding were presented in [13–16], respectively. The TV model represents another PDE-based algorithm that has been applied to image [17] and video coding [18]. In general, the TV model provides a satisfactory performance for image completion by restoring solely smooth edges. In fact, the TV model generally provides better predictions in comparison to Laplace PDE. However, its computational complexity is higher than that of the Laplace PDE algorithm.

In the following, the integration of both image completion algorithms into a hybrid block-based video codec will be further explained. Possible optimization approaches will be also described. Note that so far, PDE-based approaches have been applied only to intra prediction.

#### 5.06.3.4.1 Intra prediction based on Laplace PDE

Figure 6.2b depicts the usage of PDE for intra prediction.  $\hat{I}$  denotes the unknown area in the current block  $\Omega$ ,  $I$  is defined as the reconstructed region, and the boundary  $\partial\Omega$  represents the spatial neighbors (one sample) above and to the left of the current block. The aim is to find  $\hat{I}$  using only the information available in  $\partial\Omega$ . This boundary value problem can be expressed as:

$$\Delta \hat{I} = 0 \quad \text{with} \quad \hat{I}|_{\partial\Omega} = I|_{\partial\Omega}, \quad (6.8)$$

where  $\Delta$  represents the Laplacian operator. In this way, information on the boundary  $\partial\Omega$  is diffused into  $\Omega$ , such that the final result is smooth. Note that in the context of intra prediction, a minimum of two out of four sides of the block to-be-coded are always adjacent to the non-reconstructed area (cf. Figure 6.2b). In this case, the boundary condition (right equation in (6.8)) of the Laplace equation is incomplete. Therefore the Laplace PDE has to be adapted to use only the available samples in the considered region (cf. [15]), e.g. the spatial neighbors above and to the left of the current block as shown in Figure 6.2b. The samples in  $\hat{I}$  can be determined by solving a system of linear equations after applying the numerical

approximation of the 2D-Laplace operator [15]. A solution source code and further information can be found at [W1].

#### 5.06.3.4.2 Intra prediction based on TV

Total variation-based image completion is also a non-textured approach with the capability of restoring smooth edges. The aim of this method, in the context of intra prediction (cf. Figure 6.2b), is to determine the to-be-coded block  $\hat{I}$  from  $I$  by solving a constrained variational (minimization) problem. TV can be formalized as:

$$\nabla \cdot \left( \frac{\nabla \hat{I}}{|\nabla \hat{I}|} \right) + \lambda_e (\hat{I} - I) = 0 \quad \text{with} \quad \hat{I}|_{\partial\Omega} = I|_{\partial\Omega}, \quad (6.9)$$

where  $\lambda_e$  corresponds to the Lagrange multiplier. In a noise free situation, (6.9) is reduced to a boundary value problem [39]:

$$\nabla \cdot \left( \frac{\nabla \hat{I}}{|\nabla \hat{I}|} \right) = 0 \quad \text{with} \quad \hat{I}|_{\partial\Omega} = I|_{\partial\Omega}. \quad (6.10)$$

More details can be found in [39].

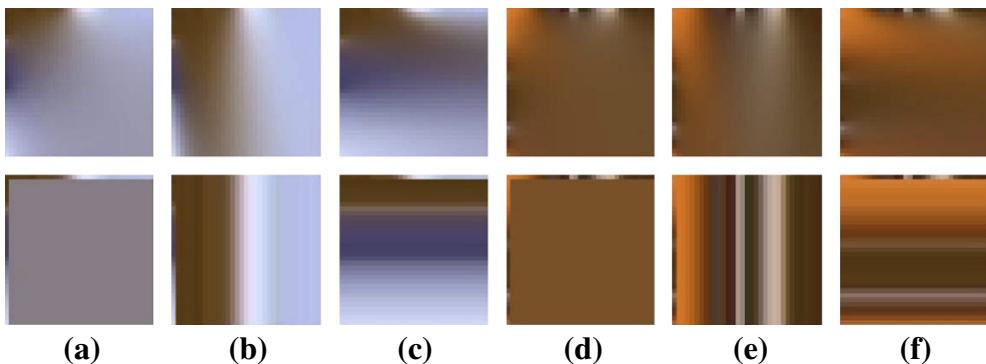
#### 5.06.3.4.3 Optimization strategies and performance

In general, Laplace image completion is suitable for predicting only smooth regions. This property of Laplace PDE restricts its application area. To avoid this limitation different strategies can be applied. For example, Doshkov et al. [15] proposed an approach to control the prediction direction of Laplace completion and minimize spatial discontinuities within blocks and their boundaries simultaneously. This is achieved by modifying the 2D-Laplace operator. Two new intra-modes (Vertical and Horizontal) are shown in Figure 6.5. A detailed mathematical explanation of all novel intra options can be found in [15] as well as in [16]. Overall, more than 2% bit rate savings have been achieved for “Foreman,” which contains long and clean edges, compared to H.264/AVC. Furthermore, exploring larger block structures in a framework similar to HEVC as proposed in [11, 16] results in average bit rate savings of about 2%. As a matter of fact, there exist also other efficient approaches that improve the Laplace performance. Bastani et al. [13] and Liu et al. [14] explicitly detect and transmit edges and use Laplace PDE only for smooth regions. However, it has to be noted that edge detection is a difficult and inaccurate operation. Therefore, in the context of video coding, this may cause other technical issues and constraints (e.g. inaccurate prediction, larger side information).

An optimization strategy of TV was proposed in the work by Qi et al. [18]. Before the to-be-coded block is predicted by the TV algorithm, a pre-prediction with Laplace PDE is applied to determine, if this block is suitable for PDE-based prediction in the rate-distortion (RD) sense. If this is the case, TV model is utilized. Bit rate reduction of up to 4% was achieved compared to HEVC (HM2.0).

#### 5.06.3.5 Non-parametric approaches

Several approaches exploit non-parametrical methods for intra [15, 19–22] and inter prediction [23–25] of texture blocks in H.264/AVC.

**FIGURE 6.5**

Comparison between PDE-based directional prediction [15] (top) and analog H.264/AVC prediction modes (bottom) with  $32 \times 32$  blocks. (a and d) Laplace PDE (top) vs. DC of H.264/AVC (bottom). (b and e) Vertical-Laplace PDE (top) vs. Vertical-H.264/AVC (bottom). (c and f) Horizontal-Laplace PDE (top) vs. Horizontal-H.264/AVC (bottom). Note, that all blocks are shown with the original spatial neighbors of one sample width (top row and outmost left column of each block).

#### 5.06.3.5.1 General template matching algorithm

The integration of the template matching algorithm for intra prediction of the current block from  $I$  is presented in [Figure 6.2c](#). This method estimates the current block by using a homogeneous Markov random fields (MRF)-based intra-synthesis approach. The MRF thereby relates to

$$p(\mathcal{P}_i | I - \{i\}) = p(\mathcal{P}_i | \mathcal{N}_i). \quad (6.11)$$

That is to the assumption that any texture pattern  $\mathcal{P}_i$  extracted from the given sample  $I$  (reconstructed area) at location  $i$  can be predicted from the corresponding neighborhood system  $\mathcal{N}_i$  and is independent of the rest of the texture. The homogeneity property presumes that the conditional probability  $p(\mathcal{P}_i | \mathcal{N}_i)$  is independent of the site  $i$  in  $I$ . MRF allows modeling of non-linear statistical dependencies between amplitude values, e.g. luminance, in a given neighborhood system [41]. Thus, using the neighboring samples above and to the left of the red marked sub-block (samples of region  $\mathcal{P}$  in [Figure 6.2c](#)), the best continuation candidate is searched in a predefined area (search area in [Figure 6.2c](#)) of  $I$ . For this, an appropriate matching criterion, e.g. Sum of Absolute Differences, is used between the templates  $\mathcal{P}$  and  $\mathcal{P}^*$ . Finally, the neighboring samples of  $\mathcal{P}^*$  are copied to the unknown area  $\Omega$ .

Note that, this texture synthesis approach assigns the output samples commonly in a raster scan order. In comparison to intra prediction, the inter predictor of a target block is generated by minimizing the matching error between the template  $\mathcal{P}$  and a search area in a previously reconstructed frame.

#### 5.06.3.5.2 Optimization strategies and performance

Different optimizations of non-parametric approaches for video coding can be found in the literature. Some of the most established methods utilize: template matching averaging, different directional

templates, priority-based template matching or a combination of the different methods. Hence, in the following, the most important techniques are briefly described.

*Template matching averaging [15,20,23,25]*: The basic template matching algorithm as described in the previous section is extended in such a way that the final candidate sub-block is the weighted average of  $n$  candidate sub-blocks with the lowest matching errors. Averaging  $n$  candidate sub-blocks results in an  $n$  times smaller variance than the individual candidates [25]. Therefore statistically, a lower prediction error can be achieved, i.e. the coding efficiency increases as less transform coefficients are required. Bit rate savings, compared to H.264/AVC, reach up to 8% for inter prediction [25] and lie above 10% for intra prediction [20].

*Directional templates [20]*: The shape of the templates is extended to exploit the directional similarity of the signal. For example, the horizontal template is constructed so that the samples of region  $P$  in Figure 6.2c occupy only the left of the target sub-block thus ignoring the contribution of the samples above during the matching procedure. It was shown in [20] that the directional templates were useful in sequences with dominant directional structures. The maximum bit rate improvements that have been reported, are higher than 6% respectively 2% for Foreman in comparison to H.264/AVC (intra prediction) and the original matching algorithm.

*Priority-based template matching [15,22]*: This method improves the prediction scheme by applying a priority-guided template matching algorithm. The proposed algorithm first calculates the priority value of each sample at  $\partial\Omega$  (cf. Figure 6.2b). Then the template matching procedure with the sub-block centered at the sample with the highest priority among all the border samples is performed. The main advantage of the priority-based template matching lies in its ability to deal with regions with mixed structure and texture. This method can save about 2% bit rate for intra prediction (cf. [22]) in comparison to the original template matching algorithm.

*Combination of different methods [15]*: This algorithm represents a two candidate templates method where a combination of both priority and original template matching is used. Firstly, the current block to-be-coded ( $\Omega$ ) is predicted separately with the two methods mentioned above. This step produces two different but perceptually similar results. Secondly, the average of both predictors is built. Applying this method for intra prediction yields more than 13% bit rate savings for Foreman and shows an average of 2% bit rate savings for large block structures ( $16 \times 16$ ) in comparison to H.264/AVC.

Furthermore, in the work of Doshkov et al. [15], it was shown that the combination of different image completion categories PDE-based and non-parametrical methods- is also very beneficial. Enabling all corresponding intra-modes yields almost 4% bit rate savings for Foreman and shows an average of 2% bit rate savings overall in the H.264/AVC framework.

#### 5.06.4 Region-based video coding techniques

In this section, region-based video coding based on texture analysis and synthesis is explained. The presented coding approach was originally proposed by the authors and will be used to illustrate essential components of analysis-synthesis-based perception-oriented coding. In addition, alternative state-of-the-art components will be presented to demonstrate the progress that has been achieved in this area.

#### 5.06.4.1 Overall video coding framework

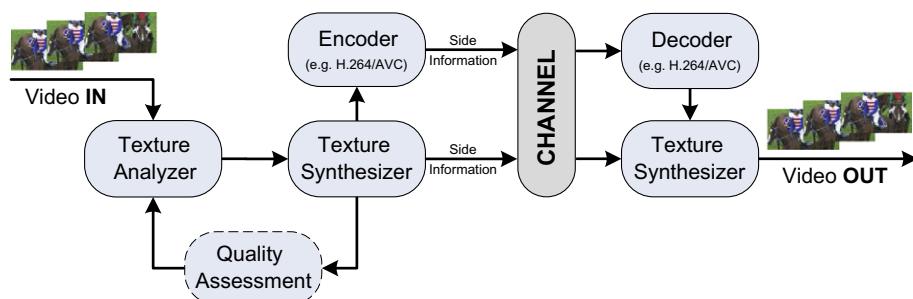
In this tutorial, the generic closed-loop video coding scheme proposed by Ndjiki-Nya et al. [30] is presented. It is assumed that many video scenes can be classified into textures that are either perceptually relevant (in an MSE sense) or perceptually irrelevant. In the TAS coding context, perceptually irrelevant textures are defined as regions with high-frequency textures. It is further assumed that for perceptually irrelevant textures, the viewer perceives the semantic meaning of the displayed texture rather than the specific details therein.

Many perceptually irrelevant texture regions are costly to encode, when using the MSE criterion as the coding distortion. Thus, in this framework, it is argued that MSE can be replaced for efficient coding of perceptually irrelevant textures. As a result, the required bit rate for transmitting perceptually irrelevant textures can be significantly reduced, if the number of bits needed for their approximate description using the modified distortion measure is smaller than the number of bits for the description using MSE.

The block diagram of the video coding scheme proposed by Ndjiki-Nya et al. [30] is depicted in Figure 6.6. As can be seen, the system comprises several sub-modules, i.e. a Texture Analyzer (TA), a Texture Synthesizer (TS), a Quality Assessor (QA) and a host encoder/decoder (e.g. H.264/AVC or HEVC) module.

Many video coding schemes based on texture analysis and synthesis can be found in the literature as explained in [4]. Most of them are open-loop, i.e. there is no mechanism to identify and where necessary alleviate artifacts due to erroneous analysis or synthesis, which yields unregulated (subjective) video quality at the decoder end. The system depicted in Figure 6.6 accounts for this major flaw in that it is a closed-loop approach. The closed-loop property is given by the fact that each synthesis path (TS → QA → TA) is repeated until the synthesis quality is validated by the QA or a maximum number of iterations has been reached.

In this approach, the incoming video sequence is evaluated via the TA module that identifies potential perceptually irrelevant textures. These textures are discriminated using color and motion features, as well as robust statistics (cf. Section 5.06.4.2). Given the perceptually irrelevant texture candidates, side information generators retrieve the required (quantized) meta data. The synthesizers used to generate synthetic versions of the original perceptually irrelevant textures utilize these data.



**FIGURE 6.6**

Typical region-based video coding architecture with TAS and optional closed-loop quality assessment.

Two synthesizers are defined that differ in the type of texture they have been optimized for. However, this is not depicted in [Figure 6.6](#) for legibility reasons.  $TS_R$  is a rigid texture synthesizer (e.g. flowers, sand), while  $TS_{NR}$  is primarily used for non-rigid texture synthesis (e.g. water, smoke). Note, that  $TS_R$  is executed first, while  $TS_{NR}$  is only executed if  $TS_R$  fails to synthesize the given texture. Although each of the texture synthesizers requires specific side information, they both have in common that they represent synthesis by example (non-parametric) approaches. That is, for each perceptually irrelevant texture to be synthesized, they require a representative texture reference for successful synthesis. Hence, the incoming video sequence is divided into Groups of Pictures (GoP, cf. [Section 5.06.4.3](#)). The synthesized (rigid or non-rigid) GoP is subsequently submitted to the video quality assessment unit for detection of possible spatial or temporal impairments in the reconstructed video. In the subsequent iterations, a state machine explores the degrees of freedom of the system for generation of relevant side information options. Once all relevant system states have been visited for the given input GoP, a rate-distortion decision is made and the optimized side information is transmitted to the decoder. Perceptually irrelevant textures for which no rate-distortion gains can be achieved are coded by the reference (host) codec, which acts as fallback coding solution.

In the following, the modules of the proposed video coding approach are explained in detail.

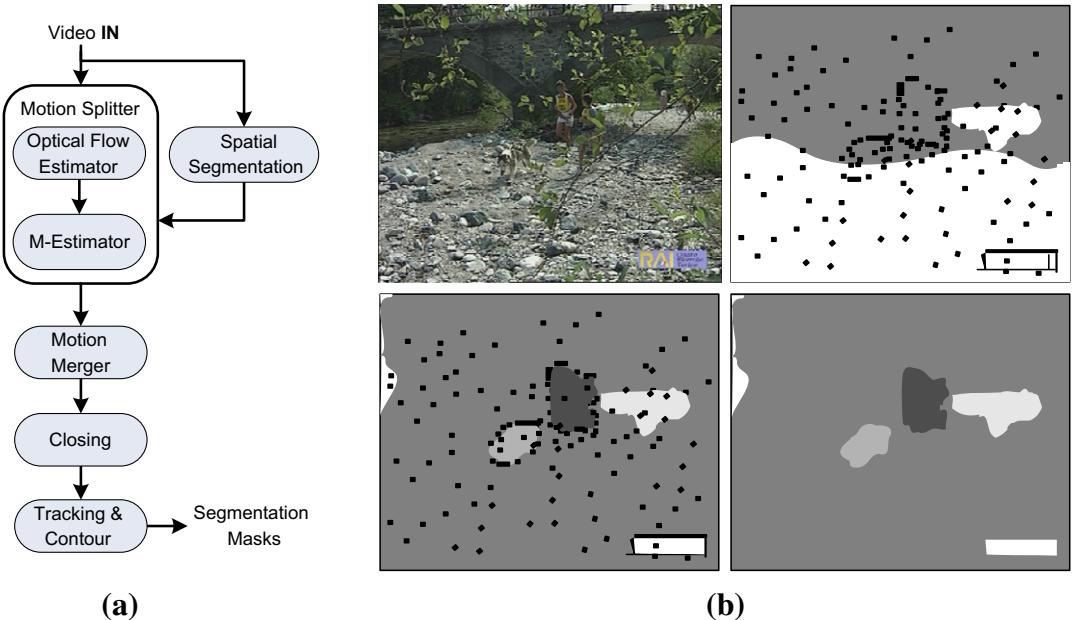
#### 5.06.4.2 Texture analysis

The main application of the texture analysis module is to identify and segment the perceptually relevant and irrelevant texture areas in a video sequence. Thus, the segmentation step is both critical and essential, as its accuracy has a significant impact on the quality of the final analysis result.

The spatio-temporal, parametric segmentation algorithm in [40], developed by the authors, is used in this framework. The principle of the algorithm is depicted in [Figure 6.7a](#). As can be seen, the proposed approach can be described as a split and merge segmentation strategy with tracking abilities. That is, at a given picture transition, the optical flow field is split into homogeneously moving regions using robust statistics, namely a maximum-likelihood estimator (M-estimator) [41]. The optical flow and subsequent M-estimation can be initialized with segmentation masks delivered by a spatial segmentation module. M-estimation is then conducted for each of the spatial regions separately. The initialization of the motion splitter module through spatial segmentation relies on the fact that good initial segments typically yield good motion estimation. Note that on the other hand, good motion estimation yields successful spatial segmentation, a classical chicken and egg dilemma.

The typically over-segmented masks obtained after the splitting step (cf. [Figure 6.7b](#), top right) are further processed by the motion merger module (cf. [Figure 6.7b](#), bottom left), which aims to convey all regions featuring similar motion properties to the same segment, which is consistent with the proposed video coding idea (cf. [Section 5.06.4.1](#)). After the merging step, a morphological closing operation is applied to remove small “holes,” i.e. small clusters with given labels, located within a much larger homogeneous texture with a different label (cf. [Figure 6.7b](#), bottom right). Finally, temporal tracking as well as contour refinement of the detected regions are performed. The output of the proposed segmentation algorithm is a mask sequence showing the location of homogeneous spatio-temporal segments in the input video sequence.

In [40] we showed that the proposed spatio-temporal segmentation algorithm has been successfully designed to meet the requirements of the TAS video coding framework. Regions found by our approach

**FIGURE 6.7**

(a) Block diagram of the proposed spatio-temporal segmentation algorithm. (b) Segmentation masks of the “Husky” test sequence (top left) before merging (top right), after merging (bottom left) and after the morphological closing operation (bottom right).

are more reliable than those found by state-of-the-art algorithms. Our algorithm moreover better detects the true object boundaries. However, our approach is prone to over-segmentation, which can be explained by a conservative merger criterion.

Although the evaluations yield an overall favorable outcome for the proposed algorithm, it cannot be ignored that some of the absolute error rates can be seen as being relatively high. For that, segmentation errors must be expected and addressed explicitly. In our video coding framework, this problem is tackled by introducing video quality assessment tools for identification of decoded video quality impairments (cp. [Section 5.06.4.4](#)).

As a matter of fact, efficient video segmentation can be designed by utilizing other tools as described in [42,43] or freely available algorithms as [W3],[W4]. Furthermore, Bosch et al. [33] evaluated the impact of different segmentation algorithms (Gray Level Co-occurrence Matrix, Gabor filters, and K-means, Split and Merge with different features, Foreground/Background Extraction) on the coding result generated with a region-based coding framework similar to the one by Ndjiki-Nya et al. as described above. The subjective experiments they conducted showed that a combination of Gray Level Co-occurrence Matrix and Split and Merge yields the better acceptance among test subjects in terms of perceived quality.

### 5.06.4.3 Texture synthesis

The texture synthesis modules used in this framework consist of two non-parametric approaches for different texture classes—rigid and non-rigid.

#### 5.06.4.3.1 Synthesis of rigid textures

The texture completion presented in this section is designed to synthesize rigid video textures. The underlying hypothesis of its conception is that rigid textures typically undergo global motion that can be explained by either camera operations or the self-motion of the corresponding foreground or background objects. Assuming that texture segmentation has generated reliable masks, the challenge then consists in modeling the global motion of the given rigid texture in a compact manner. The texture synthesizer presented in this section has a strong similarity to global motion compensation (GMC) approaches [44]. It can, however, be viewed as a generalization of these.

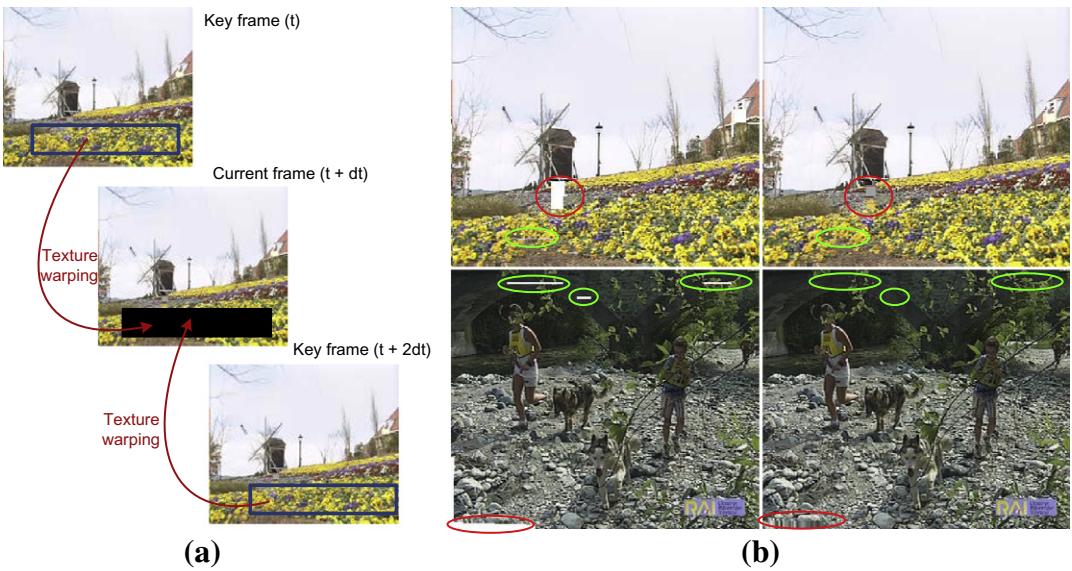
The incoming video sequence is divided into overlapping GoPs. The first GoP consists of the first picture of the sequence and the last picture of the GoP can *a priori* be set at an arbitrary distance to the first picture of the GoP. The two pictures are used as key pictures ( $K$ ) by the texture synthesizer for rigid textures. Between the key pictures, an arbitrary number of pictures, also called S pictures, are partially or fully synthesized depending on their content. For example, when 3 S pictures are used, the first GoP has the structure  $K_0SSSK_1$  in temporal order. The second GoP consists of the last picture of the first GoP (the  $K_1$  picture) and the next key picture, i.e.  $K_1SSSK_2$ . Note that the number of S pictures can in principle vary from GoP to GoP, which accounts for specific motion content in the given video sequence.

The synthesizer for rigid textures warps the missing texture from one of the key pictures toward a given synthesizable texture region identified through texture segmentation. The remaining valid, unsynthesized samples are replaced using the second reference picture (cf. Figure 6.8a). After the valid samples have been replaced, some unsynthesized samples may remain. Causes of invalidity are thereby twofold. They relate to covering and uncovering effects and to properties of the segmentation masks generated by texture segmentation. That is, samples of a detail irrelevant texture segment are considered invalid if they lie outside the corresponding region in both reference pictures. Hence, invalid samples typically occur at picture (cf. Figure 6.8b, “Husky” picture) or texture borders (cf. Figure 6.8b, “Flower Garden” and “Husky” pictures). They are tackled by using a Markov random fields (MRF) based template matching (intra only) approach (cf. Eq. (6.11)). Thus, the remaining “holes” are filled-in by assigning them corresponding color components of the defined sample that features the most similar neighborhood properties in the given picture and lies within a limited search range. A  $3 \times 3$  neighborhood and an  $11 \times 11$ -search range are used (cf. Figure 6.2c).

Intra-synthesis yields imperceptible distortions if the invalid samples represent thin line structures or “small” blobs (cf. Figure 6.8b green ellipses<sup>1</sup> in pictures). Visible distortions are however obtained for large blobs of invalid samples as can be seen in Figure 6.8b (cf. red ellipses in pictures). The annoyance of these distortions is strongly correlated with the properties of the given texture. Textures for which the stationarity is not captured by the size of the considered neighborhood may not be successfully synthesized by the template matching approach.

---

<sup>1</sup>For interpretation of color in Figure 6.8, the reader is referred to the web version of this book.

**FIGURE 6.8**

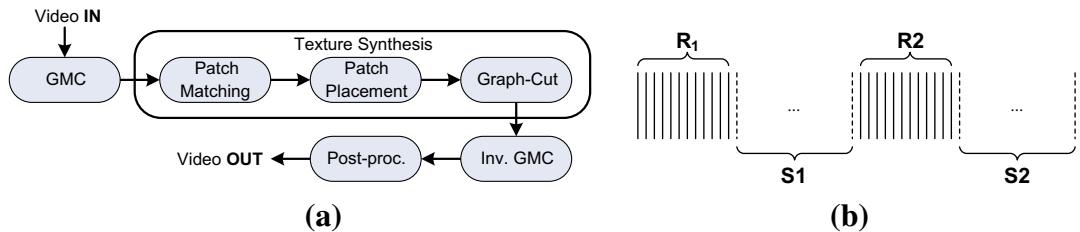
(a) Texture warping from key pictures toward a region to be filled. The missing texture is marked black in the middle picture. The warped missing area is schematically depicted in the two neighboring key pictures. (b) Success and failure of the intra-synthesis method. Picture from the “Flower Garden” test sequence with invalid samples at the texture border marked white (top left), “Flower Garden” picture with intra-synthesized invalid samples (top right), picture from the “Husky” test sequence with invalid samples at picture and texture borders marked white (bottom left), “Husky” picture with intra-synthesized invalid samples (bottom right).

Other powerful image completion approaches can be found in [4]. Also an effective joint GMC and motion segmentation method is publicly available at [W4].

#### 5.06.4.3.2 Synthesis of non-rigid textures

The texture synthesis framework, developed by Ndjiki-Nya et al. [45], is used in this coding architecture as it represents a non-parametric, template matching approach with the ability to synthesize a broad range of texture varieties. The overall structure of the approach for non-rigid textures is shown in Figure 6.9a. In the proposed scenario, the input video sequence is temporally segmented as depicted in Figure 6.9b. The first group of pictures consists of a reference burst ( $R_1$ ) that temporally precedes the synthetic burst ( $S_1$ ). The synthetic burst is itself followed by another reference burst ( $R_2$ ) in temporal order. The two reference and the synthetic bursts give a group of bursts (GoB)  $R_1S_1R_2$ . The reference bursts are chosen such that they contain the sample texture  $I$  required to synthesize the empty lattice  $\Omega$  in the synthetic burst. The second GoB consists of  $R_2S_2R_3$  and so on. Hence, an overlapping GoB structure is used.

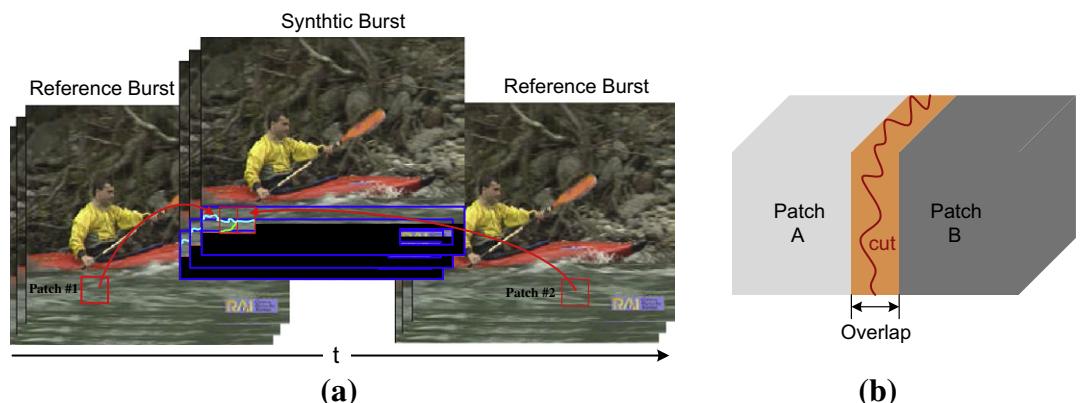
Next, a robust GMC algorithm is used to achieve a generic texture synthesis tool with regards to acceptable camera motion. GMC is based on the M-estimation algorithm presented in Section 5.06.4.2. The motion compensation algorithm consists in determining the perspective motion model [41] between adjacent pictures. Once the picture-to-picture global motion is known, each reference picture is shifted

**FIGURE 6.9**

(a) Overview of the texture synthesis algorithm by Ndjiki-Nya et al. [45]. (b) GoB structure of the texture synthesizer for non-rigid textures.

toward a given picture, e.g. the mid synthesis burst picture, by accumulation of motion parameter sets. Notice that the temporal alignment algorithm is applied on a full GoB.

Several completion approaches have been proposed in the literature [4]. Such approaches are usually limited to reconstruct relatively small holes in still images or correspond to a straightforward extension of 2D approaches, where each picture of a given video sequence is handled separately. Thus, the challenge resides in the transposition of a given texture synthesis algorithm onto an inpainting scenario, where missing textures, that can be thought of as spatio-temporal (2D + t) “holes” in a given video sequence, must be filled. This is a somewhat complicated task as both spatial and temporal inferences are required. Inappropriate synthesizer decisions may yield annoying artifacts as flickering or spurious spatio-temporal edges. Hence, in the first step, the 2D + t patch matching is executed by applying the MRF theory, i.e. an appropriate patch candidate is determined from the reference bursts (cf. Figure 6.10a). The patch placement procedure consists in filling the missing texture volume from the boundary toward the center of the hole (synthetic burst) in a helical manner. The patches considered

**FIGURE 6.10**

(a) Schematic description of the texture synthesis principle. (b) Assembly of two patches (cuboids) with a seam determined by a graph-cut algorithm.

here are cuboids (cf. Figure 6.10b). They are placed in an overlapping manner and a graph-cut [45] algorithm is applied to the overlapping region to achieve an irregular spatio-temporal boundary between the patches. This ideally decreases the perceptibility of the boundary, given an adequate cost function. The cost function is defined as follows:

$$g(l, k, \mathcal{A}, \mathcal{B}) = \frac{\|hsv_{\mathbf{a}(l)} - hsv_{\mathbf{b}(l)}\| + \|hsv_{\mathbf{a}(k)} - hsv_{\mathbf{b}(k)}\|}{\|\delta_{\mathbf{a}}(l, \vartheta)\| + \|\delta_{\mathbf{b}}(l, \vartheta)\| + \|\delta_{\mathbf{a}}(k, \vartheta)\| + \|\delta_{\mathbf{b}}(k, \vartheta)\|}, \quad (6.12)$$

where

$$\|hsv_{\mathbf{a}(\cdot)} - hsv_{\mathbf{b}(\cdot)}\| = \|h_{\mathbf{a}(\cdot)} - h_{\mathbf{b}(\cdot)}\| + \|s_{\mathbf{a}(\cdot)} - s_{\mathbf{b}(\cdot)}\| + \|v_{\mathbf{a}(\cdot)} - v_{\mathbf{b}(\cdot)}\|. \quad (6.13)$$

$\mathcal{A}$  and  $\mathcal{B}$  are two overlapping patches in the synthesis burst (cf. Figure 6.10b).  $\mathbf{a}$  corresponds to the vectorized overlapping region in patch  $\mathcal{A}$ , while  $\mathbf{b}$  is the corresponding region in patch  $\mathcal{B}$ . Furthermore,  $l$  and  $k$  are two adjacent pixels in the overlapping region, while  $\delta_{\mathbf{a}}$ ,  $\delta_{\mathbf{b}}$  represent the gradient at location  $l$  or  $k$  in direction  $\vartheta$ .  $\|\cdot\|$  represents a norm, e.g. the  $\ell_1$ -norm.  $h_{\mathbf{a}(\cdot)}$ ,  $s_{\mathbf{a}(\cdot)}$ ,  $v_{\mathbf{a}(\cdot)}$  represent the HSV components in patch  $\mathbf{a}$  at location  $l$  or  $k$ . The cost function is defined in the HSV color space due to its perceptual uniformity.

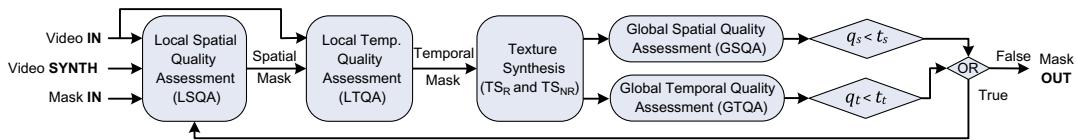
In the next step, an inverse temporal transform of the processed video sequence is operated if important global motion occurs in the video of interest. Depending on the completion scenario, the synthesized texture in the unknown area (synthetic burst) could still feature noticeable perceptual differences, especially at the transition border between the patches (cf. Figure 6.11). For this reason, a post-processing module is required to improve the perceived quality. An efficient post-processing step can be designed by utilizing tools as feathering, multiresolution spline [45] or low pass filtering at the transitions. In the work of Ndjiki-Nya et al. [46], two cloning approaches are further presented as a solution to this issue. In general, cloning corrects the reconstructed area photometrically such that subjective impairments are minimized (cf. Figure 6.11). Further results, containing videos with and without cloning post-processing, can be found at "<http://iphom.hhi.de/results/VideoSynthesisCloning.htm>." For an in-depth understanding of cloning techniques, the open source code provided at [W1] may be useful.

Of course, there exist other efficient ways to complete missing video textures. An alternative inpainting approach for object removal that exploits spatio-temporal video properties has been proposed by Patwardhan et al. [47]. A robust parametric-based framework was presented by Zhang and Bull [36].



FIGURE 6.11

Visual results for graph-cut without cloning (left image) and with cloning [46] (right image) for the (a) "Fire Fight" and (b) "Rain" sequences.

**FIGURE 6.12**

Video quality assessment method for synthesized textures by Ndjiki-Nya et al. [48].

#### 5.06.4.4 Quality assessment

The texture synthesis modules presented in the previous section generate textures that are perceptually similar to their original counterparts but objectively different. Although the texture synthesis algorithms are very efficient, they can still fail to properly synthesize a given texture class. It must thus be ensured that consistent video quality is reliably generated at the decoder end. This can be achieved by validating the output of the texture synthesis modules via objective video quality estimation measures as developed by Ndjiki-Nya et al. [48].

An example of an overall quality assessment approach for evaluation of synthetic textures is depicted in Figure 6.12 [48]. The original and synthesized sequences as well as a mask series highlighting the completed regions are input to a Local Spatial Quality Assessment (LSQA) module. The latter corrects the input mask for those areas where artifacts occur to generate a spatially modified mask (Spatial Mask). The Spatial Mask and the input videos are further input to the Local Temporal Quality Assessment (LTQA) module. LTQA modifies the Spatial Mask where applicable to generate a Temporal Mask that is used for optional texture synthesis. The global measures are then determined, i.e. the temporal (GTQA) and the spatial (GSQA) measures  $q_t$  and  $q_s$ , respectively. The obtained values are now compared to corresponding thresholds,  $t_t$  and  $t_s$ , respectively. If at least one of the former is smaller than the required quality threshold, then the whole quality evaluation is repeated. Notice that  $q_s$ ,  $q_t$  must be maximized to ensure good video quality at the decoder end. If both quality thresholds are met, then the quality evaluation process is stopped and the mask is assumed to be of satisfactory accuracy for texture synthesis. The resulting mask (Mask Out) is typically a shrunk version of the input mask generated through texture analysis.

##### 5.06.4.4.1 Global spatial quality assessment (GSQA)

The mathematical formulation of the GSQA for a single picture,  $q_s(t)$ , is given as [48]:

$$q_s(t) = \frac{\gamma}{e^{1+\delta(t)}} \quad \text{with} \quad \delta(t) = \frac{|E_o(t) - E_d(t)|}{E_o(t)}. \quad (6.14)$$

The term  $\gamma$  can be freely selected and steers the interval of  $\delta(t)$  for which the contrast is enhanced or reduced. The denominator  $e^\gamma$  is a normalization factor. The variable  $\delta(t)$  represents a differential term that assesses the distance between a given reference and a corresponding distorted signal.  $E_o(t)$  and  $E_d(t)$  correspond to mean absolute spatial gradients in the original and the distorted signal respectively.  $\delta(t)$  typically features values that lie in the interval  $[0, 1]$ , which is however not guaranteed.

$q_s(t)$  inherently detects blurring artifacts, when high frequency areas, that are particularly affected by lowpass effects, are analyzed. Furthermore, if  $E_d(t)$  is larger than  $E_o(t)$ , this indicates that the distorted signal has gained some additional edges compared to the original picture. This may be related to tiling effects in the distorted signal. On the other hand, if  $E_d(t)$  is smaller than  $E_o(t)$ , it can be assumed that a loss of contrast has occurred between original and distorted signals.

#### 5.06.4.4.2 Global temporal quality assessment (GTQA)

The proposed temporal video quality measure consists of evaluating the motion properties of the synthetic textures by matching them with the motion properties of their original counterparts. This is carried out by detecting possible motion inconsistencies based on temporal gradients. For that,  $x - t$  and  $y - t$  slices are defined to assess complex motion. As can be seen in Figure 6.13a, these slices are determined by considering one of the spatial coordinates to be fixed, i.e. the component in the  $y - t$  case and the  $y$  component in the  $x - t$  constellation. The slice definition can be formalized as follows:

$$s'_{y,o}(x, y, t, \beta) = s_{y,o}(x, y, t) * f_\beta(x, y) \quad s'_{x,o}(x, y, t, \beta) = s_{x,o}(x, y, t) * f_\beta(x, y), \quad (6.15)$$

where  $s_{x,o}(x, y, t)$  and  $s_{y,o}(x, y, t)$  represent the  $x - t$  and  $y - t$  slices in the original video signal respectively.  $s_{x,d}(x, y, t)$  and  $s_{y,d}(x, y, t)$  can be interpreted correspondingly for the distorted video.  $f_\beta(x, y)$  is a linear, anisotropic gradient filter (e.g. Sobel) of orientation  $\beta$  and “ $*$ ” represents the convolution operation. Hence,  $s'_{y,o}(x, y, t, \beta)$  and  $s'_{x,o}(x, y, t, \beta)$  correspond to highpass filtered original slices. Figure 6.13b depicts merged  $y - t$  and  $x - t$  slices of the original “Canoe” video sequence for the two main diagonal orientations ( $\beta \in \{45^\circ, 135^\circ\}$ ). It can be seen that salient locations that feature higher motion activity are brighter than such with less motion. The directional global motion activity

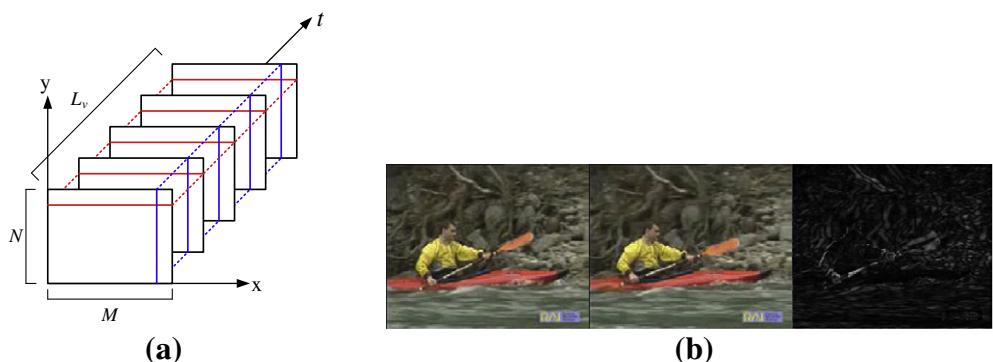


FIGURE 6.13

(a) “ $x - t$ ” (red) and “ $y - t$ ” (blue) slices in a video sequence. (b) Temporal filtering example (right) of two consecutive pictures (left and middle) of the “Canoe” video sequence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this book.)

at a given picture transition  $\Delta t$ , can be defined as

$$\phi_x^o(\Delta t, \beta) = \frac{1}{K} \sum_{x=1}^M \sum_{y=1}^N |s'_{x,o}(x, y, \Delta t, \beta)| m(x, y, \Delta t), \quad (6.16)$$

$m(x, y, \Delta t)$  represents a binary mask that is set at locations of interest and constitutes the union of the considered picture pair.  $K$  is the size of the locations of interest in the binary mask with  $K \leq MN$ .  $\phi_y^o(\Delta t, \beta)$  can be easily derived from (6.16).

The overall global motion information ( $\phi_x^o(\Delta t)$ ) for a given time interval  $\Delta t$  and slice type can now be formulated as the sum of all considered edge orientations  $\beta$ . The distorted (synthesized) signal is determined correspondingly. The temporal quality measure for a given time interval  $\Delta t$  and slice type can now be determined as

$$q_t^x(\Delta t) = \frac{\frac{\gamma}{e^{1+\delta_x(\Delta t)}}}{e^\gamma} \quad \text{with} \quad \delta_x(\Delta t) = \frac{|\phi_x^o(\Delta t) - \phi_x^d(\Delta t)|}{\phi_x^o(\Delta t)}, \quad (6.17)$$

$q_t^y(\Delta t)$  and  $\delta_y(\Delta t)$  can be easily derived from (6.17). The slice independent temporal quality measure  $q_t(\Delta t)$  can then be obtained as:

$$q_t(\Delta t) = \min [q_t^x(\Delta t), q_t^y(\Delta t)]. \quad (6.18)$$

That implies that the slice type with the largest distortions is selected for the overall quality measurement.

#### 5.06.4.4.3 Local Spatial Quality Assessment (LSQA)

Local spatial artifact detection is conducted by applying the global spatial measure (6.14) to each pixel location  $(x, y, t)$  in a given region-of-interest defined by  $m(x, y, t)$ . This can be formalized as simple replacement of “ $(t)$ ” by “ $(x, y, t)$ ” in (6.14). Note that here, the synthetic textures are evaluated only at the borders of the region-of-interest, which relates to the fact that spatial impairments due to texture synthesis typically occur at transitions between original and synthetic textures in the shape of spurious edges. Once all the  $q_s(x, y, t)$  in the transition area have been computed, suspect locations are determined and marked perceptually relevant, as they cannot be suitably reconstructed via texture synthesis. Hence the mask showing perceptually irrelevant regions is shrunk. The latter are defined as LSQA values (6.14) lower than a given threshold.

#### 5.06.4.4.4 Local Temporal Quality Assessment (LTQA)

Local temporal artifact detection is conducted by applying the global temporal measure (6.18) to single pixel locations  $(x, y, t)$  in a given region-of-interest. That means that we replace “ $(\Delta t)$ ” by “ $(x, y, \Delta t)$ ” in (6.17). Once  $q_t^x(x, y, \Delta t)$  and  $q_t^y(x, y, \Delta t)$  have been computed for a given picture pair, suspect locations are determined as local quality measure values lower than a given threshold. The critical size of clusters of such locations to be seen as potentially erroneous constitutes a degree of freedom of the approach.

The local impairment predictors (spatial and temporal) can be applied to rigid and non-rigid texture synthesizers. It must however be noticed that the mask  $m(x, y, \Delta t)$  is initialized differently in both cases. The fundamental difference between the two synthesizers resides in the fact that the synthesizer

**FIGURE 6.14**

Mask correction through local video quality assessment for (a) “Concrete” and (b) “Flower Garden.” Texture analyzer mask (left), updated mask (right).

for rigid textures computes each synthesized picture independently of the others, while the synthesizer for non-rigid textures links subsequent pictures by optimizing their textures simultaneously (cf. Section 5.06.4.3). Hence, the mask for non-rigid texture synthesis is obtained by computing the union of all masks in the considered group of bursts.

The influence of the local video quality assessment measures on the outcome of the segmentation mask is shown in Figure 6.14, where the red areas correspond to detail-irrelevant textures. The picture on the left hand side corresponds to the mask obtained from the texture analyzer, while the right picture is the mask after correction by the local video quality measures. It can be seen that in both cases the amount of segmentation mistakes is significantly reduced, i.e. the mean synthetic area is corrected from 62% to 52% for “Concrete” and from 54% to 34% for “Flower Garden.”

#### **5.06.4.4.5 Challenges and alternative quality assessment measures**

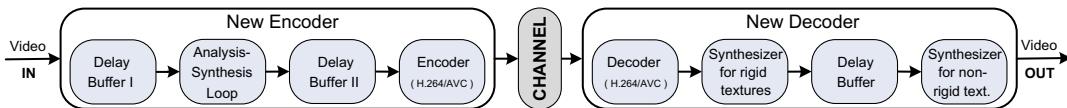
The proposed quality assessment approaches evidence that the main quality measurement challenges can be formulated as follows [4]:

- *Texture quality:* The required measures should be able to predict the perceived quality/similarity of synthesized but perceptual similar regions.
- *Localized distortions:* The required measures should be able to determine the effects of strongly localized distortions (spatial, temporal, and spatio-temporal) on the perceived overall video quality.
- *Spatio-temporal pooling:* The required measures should be able to integrate spatio-temporal information.

There exist alternative perceptual-based video quality estimation measures (e.g. SSIM, STSIM, CW-SSIM—cf. [4]) that can be applied instead of the above presented approach. An extensive overview and discussion on the most relevant (recently proposed) perceptual-based video quality assessment methods can be found in [4].

#### **5.06.4.5 System integration and performances**

The system integration consists in the integration of the proposed analysis-synthesis loop into the H.264/AVC encoder and the incorporation of the texture synthesizers into the H.264/AVC decoder

**FIGURE 6.15**

Integration of the analysis-synthesis method into H.264/AVC video codec.

(cf. Figure 6.15). The incorporation of texture segmentation into the encoder requires delay buffers to conduct a look-ahead video evaluation in order to operate a synthesis mode decision. The first delay buffer collects an entire GoP before the analysis-synthesis loop is called. Thorough content analysis is then conducted. A second delay buffer retains the incoming GoPs until the required amount of pictures is available for coding. The length of the second buffer depends on the texture properties and is basically dictated by the texture synthesizer for non-rigid textures. Once the required amount of pictures is reached, one GoP is coded.

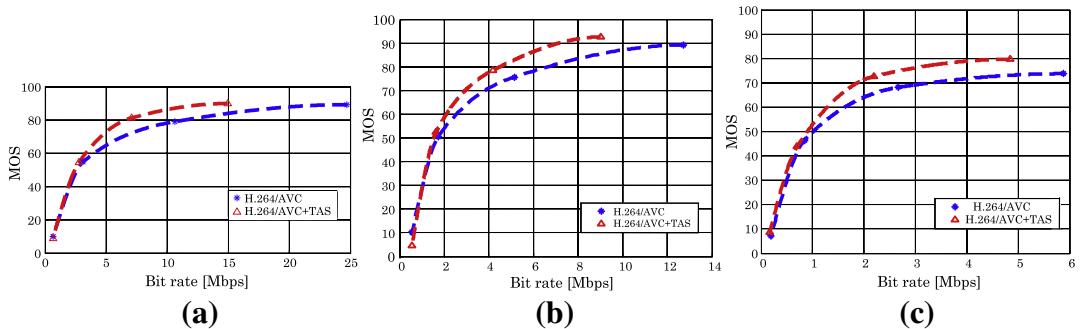
For efficient coding, different side information sets and the default—H.264/AVC—coding modes are considered. In general, all blocks belonging to a synthesizable texture region are handled as skipped blocks (cf. [1]). While decoding, the bit-stream is firstly decoded by the standard-conforming decoder (cf. Figure 6.15). Texture synthesis for rigid textures is then operated. Output pictures of the rigid texture synthesizer are stored in a delay buffer if at least one non-rigid texture is to be synthesized. Once all pictures of the corresponding reference and synthesis bursts are released, non-rigid texture synthesis is carried out.

#### **5.06.4.5.1 System configuration**

For the H.264/AVC codec, one reference picture for each P picture, CABAC (entropy coding method), rate-distortion optimization and 30 fps are set. For that, three hierarchically structured B pictures are used in the experiments:  $IB_2B_1B_2PB_2B_1B_2P\cdots P$  and I pictures are coded with QP + 1, while the  $B_1$  pictures are coded with QP + 5 and the  $B_2$  pictures with QP + 6.

A search range of  $18 \times 5$  (abscissa  $\times$  ordinate) is used for  $TS_R$  at CIF ( $352 \times 288$ ) resolution, while a range of  $36 \times 10$  is used for SD ( $720 \times 576$ ) resolutions for motion estimation. Reference bursts are assigned a length of 20 pictures, while synthetic bursts have a length of 70 pictures. The size of the Delay Buffer II (cf. Figure 6.15) is thus set to 110 pictures in our experiments. For the video quality assessor, following thresholds are used: A cluster of 12 pixels is required for a synthetic texture neighborhood to be considered as suspect for LTQA (cf. Figure 6.12). The threshold for LSQA is set to 0.15, while the global thresholds are set to 72 and 54 for spatial (GSQA) and temporal (GTQA) artifacts respectively (cf. Figure 6.12).

The proposed coding scheme is evaluated for three test sequences (“Flower Garden,” “Concrete,” and “Sea”) containing rigid and non-rigid textures. Furthermore, subjective tests have been conducted to achieve a rate-quality curve, which is more likely to provide in-depth insight into the potentialities of the proposed approach. The triple stimulus continuous evaluation scale (TSCES) method [49] was used for subjective evaluations. 31 test subjects (non-experts) were asked to evaluate the performance of the codec system. “Flower Garden” and “Concrete” were shown at SD resolution, while “Sea” was shown at CIF resolution. For each sequence, four QP levels (16, 24, 32, and 40) were evaluated.

**FIGURE 6.16**

Rate-quality curves for (a) “Concrete,” (b) “Flower Garden,” and (c) “Sea.” Genuine H.264/AVC (red curves) vs. H.264/AVC with texture analysis and synthesis (blue curves). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this book.)

#### 5.06.4.5.2 Rate-quality performance

Rate-quality curves for the “Concrete” video sequence are shown in Figure 6.16a. It can be seen that the curve corresponding to the proposed method (TAS, red curve with triangles) has better characteristics than the H.264/AVC curve (blue curve with asterisks). For QP 32 and QP 40, both codecs yield similar performances. Such that the measured differences between the codecs cannot be considered as statistically relevant. However, at higher bit rates, statistically relevant deviations are observed between the curves. For “Concrete,” the subjective bit rate gain achieved by the proposed codec can be given as 41% at QP 16 and 39% at QP 24. Similar observations can be made for the other test sequences. For “Flower Garden” (cf. Figure 6.16b) the subjective bit rate gain is determined as 40% at QP 16 and 24% at QP 24. Finally, for “Sea” (cf. Figure 6.16c), the subjective bit rate gain is given as 17.6% at QP 16 and 18.5% at QP 24. It should however be noted that the bit rate gains occur at quite high bit rates that are usually not affordable in practice. This will be the subject of future work.

## 5.06.5 Conclusion

This chapter presented emerging video compression approaches based on image analysis and synthesis. A range of block-based and region-based coding strategies have been described including new and efficient approaches proposed by the authors. The potential of these methods has been addressed when integrated into hybrid block-based coding frameworks. Also, their performances and alternative optimization possibilities were presented. The experimental results showed that TAS techniques can improve the coding efficiency of H.264/AVC and HEVC intra and inter prediction by up to 18% and 11%, respectively.

Furthermore, the whole structure of an automatic region-based video coding approach was presented. Consistent quality of the decoded video signal is ensured by operating closed-loop analysis-synthesis with incorporated objective measurement of subjective quality. The experiments showed

that the presented algorithm yields bit rate savings of up to 41% compared to a standard conforming H.264/AVC video codec.

---

## Relevant websites of open source software

- [W1] Smooth Image Completion: <http://www.leyvand.com/research/adv-graphics/ex1.htm#guidelines>
  - [W2] TV Image Reconstruction: <http://www2.imm.dtu.dk/~pch/mxTV/>
  - [W3] Video Segmentation: [http://www.cse.buffalo.edu/~jcorso/r/snippets.video\\_segmentation.html](http://www.cse.buffalo.edu/~jcorso/r/snippets.video_segmentation.html)
  - [W4] Parametric Image Completion: [http://lcav.epfl.ch/reproducible\\_research/CostantiniIP07\\_1](http://lcav.epfl.ch/reproducible_research/CostantiniIP07_1)
- 

## Glossary

<b>Artifact</b>	an inaccurate observation, effect, or result, especially one resulting from the technology used in scientific investigation or from experimental error
<b>Bit stream</b>	a sequence of bits that forms the representation of coded pictures and associated data forming one or more coded video sequences
<b>Data element</b>	the fundamental data structure in a data processing system
<b>Decoder</b>	an embodiment of a decoding process that reads a <i>bit stream</i> and derives decoded pictures from it
<b>Encoder</b>	an embodiment of an encoding process that produces a <i>bit stream</i> conforming to a given specification, e.g. an International Standard
<b>Image/texture completion</b>	generalized definition of all approaches known as <i>inpainting</i> , <i>texture synthesis</i> or image/video restoration
<b>Inpainting</b>	refers to the regeneration process of missing or damaged image areas using information from their vicinity
<b>Intra prediction</b>	a <i>prediction</i> derived from only <i>data elements</i> (e.g. sample value) of the same decoded picture (slice)
<b>Inter prediction</b>	a <i>prediction</i> derived from only <i>data elements</i> (e.g. sample value or motion vector) of reference pictures other than the current decoded picture
<b>Motion vector</b>	a two-dimensional vector used for <i>inter prediction</i> that provides an offset from the coordinates in the decoded picture to the coordinates in a reference picture
<b>Non-rigid (dynamic) textures</b>	texture type that exhibits local motion and ideally temporal stationarity, e.g. video sequences containing water surfaces, clouds, fire, whirlwind, etc.
<b>Prediction</b>	an embodiment of the prediction process that uses a combination of specified values or previously decoded <i>data elements</i> (e.g. sample value or motion vector) to provide an estimate of the <i>data element</i> currently being decoded
<b>Residual</b>	the decoded difference between a <i>prediction</i> of a sample or <i>data element</i> and its original value
<b>Rigid (static) textures</b>	spatial texture type that ideally exhibits spatial stationarity, e.g. wall, grass, stone surfaces, etc.

<b>Template matching</b>	the process of finding small segments of an image/video which match a template segment
<b>Texture analysis</b>	refers to a class of image processing approaches or models that characterize the spatial or temporal properties of textures
<b>Texture segmentation</b>	the process of partitioning an image into multiple segments (homogeneous texture regions)
<b>Texture synthesis</b>	refers to the generation process of a novel texture pattern from a limited sample set

---

## References

- [1] ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Advanced Video Coding for Generic Audiovisual Services, v1, May 2003, v2, January 2004, v3 (with FRAExt), September 2004, v4, July 2005.
- [2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G.J. Sullivan, Rate-constrained coder control and comparison of video coding standards, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 688–703.
- [3] G.J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Trans. Circ. Syst. Video Technol.* (2012).
- [4] P. Ndjiki-Nya, D. Doshkov, H. Kaprykowsky, F. Zhang, D. Bull, T. Wiegand, Perception-oriented video coding based on image analysis and completion: a review, *Signal Process. Image Commun.* 27 (6) (2012) 579–594.
- [5] A. Khandelia, S. Gorecha, B. Lall, S. Chaudhury, M. Mathur, Parametric video compression scheme using AR based texture synthesis, in: Proceedings of ICCV, Graphics and Image Processing, Bhubaneswar, India, December 2008, pp. 219–225.
- [6] J. Ballé, A. Stojanovic, J.-R. Ohm, Models for static and dynamic texture synthesis in image and video compression, *IEEE J. Sel. Top. Signal Process.* 5 (7) (2011) 1353–1365.
- [7] H. Chen, R. Hu, D. Mao, R. Thong, Z. Wang, Video coding using dynamic texture synthesis, in: Proceedings of ICME, Singapore, July 2010, pp. 203–208.
- [8] L. Liu, Y. Liu, E.J. Delp, Content-adaptive motion estimation for efficient video compression, in: Proceedings of SPIE: International Conference on Visual Communications and Image Processing, San Jose, California, 2007.
- [9] L. Liu, Y. Liu, E.J. Delp, Enhanced intra prediction using context-adaptive linear prediction, in: Proceedings of PCS Picture Coding Symposium, 2007.
- [10] J. Chen, W. Han, Adaptive linear prediction for block-based lossy image coding, in: Proceedings of ICIP, Cairo, Egypt, 2009, pp. 2833–2836.
- [11] D. Doshkov, O. Jottrand, T. Wiegand, P. Ndjiki-Nya, On the efficiency of image completion methods for intra prediction in video coding with large block structures, in: Proceedings of SPIE: Visual Information Processing and Communication IV, San Francisco, California, USA, 2013.
- [12] D.C. Garcia, R.L. Queiroz, Least-squares directional intra prediction in H.264/AVC, *IEEE Signal Process. Lett.* 17 (10) (2010) 831–834.
- [13] V. Bastani, M.S. Helfroush, K. Kasiri, *Image Compression Based on Spatial Redundancy Removal and Image Inpainting*, Zhejiang University Press co-published with Springer, January 2010, pp. 91–100.
- [14] D. Liu, X. Sun, F. Wu, Y.-Q. Zhang, Edge-oriented uniform intra prediction, *IEEE TIP* 17 (10) (2008) 1827–1836.
- [15] D. Doshkov, P. Ndjiki-Nya, H. Lakshman, M. Köppel, T. Wiegand, Towards efficient intra prediction based on image inpainting methods, in: Proceedings of PCS, Nagoya, Japan, 2010, pp. 470–473.

- [16] H. Kaprykowsky, D. Doshkov, C. Hoffmann, P. Ndjiki-Nya, T. Wiegand, Investigation of perception-oriented coding techniques for video compression based on large block structures, Proceedings of SPIE: Applications of Digital Image Processing XXXIV, San Diego, California, USA, vol. 8135, 2011, pp. 81350K1–81350K15.
- [17] J. Xu, J. Ma, D. Zhang, Y. Zhang, S. Lin, Improved total variation minimization method for compressive sensing in intra-prediction, *Signal Process. Image Commun.* 92 (11) (2012) 2614–2623.
- [18] X. Qi, T. Zhang, F. Ye, A. Men, B. Yang, Intra prediction with enhanced inpainting method and vector predictor for HEVC, in: Proceedings of ICASSP, 2012, pp. 1217–1220.
- [19] T.K. Tan, C.S. Boon, Y. Suzuki, Intra prediction by template matching, in: Proceedings of ICIP, Atlanta, GA, USA, 2006, pp. 1693–1696.
- [20] T.K. Tan, C.S. Boon, Y. Suzuki, Intra prediction by averaged template matching predictors, in: Proceedings of CCNC, Las Vegas, Nevada, USA, January 2007, pp. 405–409.
- [21] J. Ballé, M. Wien, Extended texture prediction for H.264/AVC intra coding, in: Proceedings of ICIP, San Antonio, Texas, USA, September 2007, pp. 93–96.
- [22] Y. Guo, Y.-K. Wang, H. Li, Priority-based template matching intra prediction, in: Proceedings of ICME, Hannover, Germany, June 2008, pp. 1117–1120.
- [23] K. Sugimoto, M. Kobayashi, Y. Suzuki, S. Kato, C.S. Boon, Inter frame coding with template matching spatio-temporal prediction, in: Proceedings of ICIP, Singapore, October 2004, pp. 465–468.
- [24] M. Kobayashi, Y. Suzuki, C.S. Boon, T. Horikoshi, Reduction of information with motion prediction using template matching, Proceedings of PCS, Japan, November 2005, pp. 17–18.
- [25] Y. Suzuki, C.S. Boon, C.S. Tan, Inter frame coding with template matching averaging, in: Proceedings of ICIP, San Antonio, Texas, USA, September 2007, pp. 409–412.
- [26] C. Zhu, X. Sun, F. Wu, H. Li, Video coding with spatio-temporal texture synthesis, in: Proceedings of ICME, Beijing, China, July 2007, pp. 112–115.
- [27] J.Y.A. Wang, E.H. Adelson, Representing moving images with layers, *IEEE Trans. Image Process. Spec. Issue Image Seq. Compress.* 3 (5) (1994) 625–638.
- [28] S.-Y. Yoon, E.H. Adelson, Subband texture synthesis for image coding, Proceedings of SPIE on Human Vision and Electronic Imaging III, San Jose, CA, USA, vol. 3299, January 1998, pp. 489–497.
- [29] A. Dumitraş, B.G. Haskell, An encoder-decoder texture replacement method with application to content-based movie coding, *IEEE Trans. Circ. Syst. Video Technol.* 14 (6) (2004) 825–840.
- [30] P. Ndjiki-Nya, C. Stüber, T. Wiegand, Generic and robust video coding with texture analysis and synthesis, in: Proceedings of IEEE ICME, Beijing, China, 2007.
- [31] C. Zhu, X. Sun, F. Wu, H. Li, Video coding with spatio-temporal texture synthesis and edge-based inpainting, in: Proceedings of ICME, Hannover, Germany, June 2008, pp. 813–816.
- [32] M. Bosch, F. Zhu, E. Delp, Spatial texture models for video compression, Proceedings of ICIP, San Antonio, TX, USA, vol. 1, September 2007, pp. 93–96.
- [33] M. Bosch, F. Zhu, E.J. Delp, Perceptual quality evaluation for texture and motion based video coding, in: Proceedings of ICIP, Cairo, Egypt, November 2009.
- [34] B.T. Oh, Y. Su, A. Segall, C.-C.J. Kuo, Synthesis-based texture coding for video compression with side information, in: Proceedings of ICIP, San Diego, CA, USA, October 2008, pp. 1628–1631.
- [35] F. Zhang, D.R. Bull, Enhanced video compression with region-based texture models, in: Proceedings of PCS, Nagoya, Japan, December 2010, pp. 54–57.
- [36] F. Zhang, D.R. Bull, A parametric framework for video compression using region-based texture models, *IEEE J. Sel. Top. Signal Process.* 5 (7) (2011) 1378–1392.
- [37] M.O. Szummer, Temporal texture modeling (Master thesis), Massachusetts Institute of Technology, USA, September 1995.
- [38] G. Doretto, A. Chiuso, Y.N. Wu, S. Soatto, Dynamic textures, *Int. J. Comput. Vis.* 51 (2) (2003) 91–109.

- [39] T.F. Chan, J. Shen, Mathematical models for local non-texture inpaintings, *SIAM J. Appl. Math.* (2001) 1019–1043.
- [40] P. Ndjiki-Nya, S. Gerke, T. Wiegand, Improved video segmentation through robust statistics and MPEG-7 features, in: Proceedings of ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing, Taiwan, April 2009.
- [41] J.-R. Ohm, *Multimedia Communication Technology*, Springer, Berlin, Heidelberg, New York, 2004, ISBN: 3-540-01249-4.
- [42] C.C. Dorea, M. Pardàs, F. Marqués, A motion-based binary partition tree approach to video object segmentation, in: Proceedings of ICIP IEEE International Conference on Image Processing 2005, Genova, Italy, vol. 2, September 2005, pp. 430–433.
- [43] R. O’Callaghan, D. Bull, Combined morphological spectral unsupervised image segmentation, *IEEE Trans. Image Process.* 14 (1) (2005) 49–62.
- [44] A. Smolić, *Globale Bewegungsbeschreibung und Video Mosaiking unter Verwendung parametrischer 2-D Modelle, Schätzverfahren und Anwendungen* (Ph.D. thesis), Aachen University of Technology, Germany, 2001.
- [45] P. Ndjiki-Nya, C. Stüber, T. Wiegand, Texture synthesis method for generic video sequences, in: Proceedings of ICIP, San Antonio, Texas, USA, September 2007, pp. 397–400.
- [46] P. Ndjiki-Nya, D. Doshkov, M. Köppel, T. Wiegand, Optimization of video synthesis by means of cost-guided multimodal photometric correction, in: Proceedings of the Sixth International Symposium on Image and Signal Processing and Analysis (ISPA), Salzburg, Austria, 2009.
- [47] K.A. Patwardhan, G. Sapiro, M. Bertalmio, Video inpainting under constrained camera motion, *IEEE Trans. Image Process.* 16 (2) (2007) 545–553.
- [48] P. Ndjiki-Nya, M. Barrado, T. Wiegand, Efficient full-reference assessment of image and video quality, in: Proceedings of IEEE ICIP, San Antonio, TX, USA, September 2007.
- [49] H. Hoffmann, HDTV – EBU format comparisons at IBC 2006, *EBU Technical Review*, October 2006, pp. 1–8.

# Measuring Video Quality

# 7

Fan Zhang and David R. Bull

*Bristol Vision Institute, University of Bristol, Bristol BS8 1UB, UK*

---

## Nomenclature

ACR	Absolute Category Rating
AVM	Artifact-Based Video Metric
DMOS	Difference Mean Opinion Score
DSCQS	Double Stimulus Continuous Quality Scale
DSIS	Double stimulus impairment scale
GOP	Group of pictures
HVS	Human Visual System
LCC	Linear Correlation Coefficient
MAD	Most Apparent Distortion
ME	Motion Estimation
MOS	Mean Opinion Score
MOVIE	Motion-based video integrity evaluation
OR	Outlier ratio
PSNR	Peak signal-to-noise ratio
PVM	Perception-based video metric
RDO	Rate-distortion optimization
RMSE	Root Mean Squared Error
RQO	Rate-quality optimization
SAMVIQ	Subjective Assessment Methodology for Video Quality
SI	Spatial information
SROCC	Spearman Rank Order Correlation Coefficient
SSIM	Structural Similarity Image Metric
STMAD	Spatio-temporal most apparent distortion
TI	Temporal Information
TSCES	Triple Stimulus Continuous Evaluation Scale
VQEG	Video Quality Experts Group
VQM	Video Quality Model
VSNR	Visual Signal-to-Noise Ratio

### 5.07.1 Introduction

Assessing the perceptual quality of video content is a basic and important requirement of image and video processing. Visual perception is highly complex, influenced by many confounding factors, not fully understood and difficult to model. Assessment of the quality of moving images is even more complicated than that for still image since, apart from spatial information associated with 2-D images, temporal aspects of the source also affects the overall perceptual quality. For these reasons, subjective assessments are still frequently used as a benchmark for video quality, where a group of human viewers are asked their opinions on quality under a range of test conditions.

Objective measures of video quality have conventionally been computed using the absolute or squared difference between the distorted version of frames and their reference version. It is however well known that the perceptual distortion experienced by the human viewer cannot be fully characterized using such simple mathematical differences. Because of the limitations of these distortion-based measures, perception-based metrics have begun to replace them. These offer the potential for enhanced correlation with subjective opinions, thus enabling more accurate estimates of visual quality.

This chapter first introduces the aim of video quality assessment and describes the factors that influence our perception of quality. Next, in [Section 5.07.3](#), we review common methods used for conducting subjective trials and for analyzing the results from them. Based on these, a discussion on the properties of some publicly available subjective test databases is given in [Section 5.07.4](#). These databases provide an important benchmark for the development and validation of efficient perception-based video quality metrics. The most promising objective video quality metrics are reviewed in [Section 5.07.5](#) and a performance comparison is provided. Finally, in [Section 5.07.6](#), we conclude the chapter and highlight the future work needed in this area.

---

### 5.07.2 Background

The primary motivations for measuring video quality include: (i) comparing the performance of different video codecs across a range of bit rates and content types; (ii) comparing the performance of different video codecs across a range of channel impairments; (iii) comparing the influence of various parameters and coding options for a given type of codec. The latter is of particular relevance to in-loop Rate-Distortion (or Quality) Optimization RDO (RQO).

Good overviews of the reasons and processes for evaluating visual quality are provided in Refs. [1,2].

#### 5.07.2.1 General approaches to measuring video quality

Video quality can be assessed using either subjective or objective methods.

Subjective quality assessment requires many observers and many presentations of a representative range of impairment conditions and content types. Subjective testing conditions must be closely controlled, with appropriate screening of observers and post-processing of the results to ensure consistency and statistical significance. They are costly and time consuming, but generally effective.

Objective quality metric uses metrics that attempt to capture the perceptual mechanisms of the human visual system (HVS). The main issue here is that simple metrics bear little relevance to the HVS and generally do not correlate well with subjective results, especially at lower bit rates when distortions are higher. More complex, perceptually inspired metrics, although improving significantly in recent

**Table 7.1** Influential Factors for Visual Perception

Viewing environment	Content types	Artifact types
Display parameters	Luminance levels	Edges
Ambient lighting	Textures	Blurring
Viewing distance	Region of interest	Ringing
Audio quality	Motion	Dissimilarity

NB: Display parameters include display size, brightness, dynamic range, and resolution.

years, can still be inconsistent under certain test conditions. The outcome of this is that mean squared error (MSE) based metrics are still frequently employed, both for in-loop optimization and for external performance comparisons.

### 5.07.2.2 Influential factors on video quality perception

More than a century ago, vision scientists began to pay attention to our perceptual sensitivity to image and video distortions. This sensitivity varies with screen brightness [3–5], local spatial and temporal frequency characteristics [6], types of motion, eye movements, various types of artifact and, of course, with the viewing environment [1]. These factors are summarized in **Table 7.1**. In order to ensure validity of subjective tests and consistency in the performance of objective metrics, the influence of these sensitivities must be, as far as possible, represented in the content used for evaluations and captured in the structure of any metric employed.

It should also be noted that the performance of the HVS varies significantly across subjects, depending on age, illness, fatigue, or visual system defects. It is also possible that biases can influence the opinions of viewers through personal preferences or boredom.

In order to provide consistent subjective assessments of video quality, it is essential that the details and parameters of the viewing conditions are recorded and kept as consistent as possible between tests. This is particularly important when comparing results across different laboratories.

---

## 5.07.3 Subjective testing

Despite recent advances in the performance of objective metrics, none are yet universally accepted as a definitive measure of quality.<sup>1</sup> As a consequence it is necessary to use controlled subjective testing. Subjective assessment methods are employed widely to characterize and compare or validate the performance of video compression algorithms. Based on a representative set of test content and impairment conditions, they can provide a robust indication of the reactions of the typical viewer.

A general methodology for subjective testing is described as follows.

- (i) Select the codecs under test and the parameter settings to be used. (GOP structures, filtering, RDO modes, ME modes, etc.)

---

<sup>1</sup>It could be argued that VQM [7], as standardized by ANSI, is more widely accepted than others although it is not the best performing.

- (ii) Define the coding conditions to be evaluated. (spatial and temporal resolutions, bit rates, etc.)
- (iii) Identify or acquire the test sequences that will be used, ensuring that they are sufficiently varied and challenging, representative of the applications of interest and of appropriate duration.
- (iv) Select an appropriate evaluation methodology. (dependent on conditions being tested, impairment levels, etc.)
- (v) Design the test session and environment (complying with recommendations of whichever methodology is being adopted).
- (vi) Dry run the session to remove any bugs or inconsistencies.
- (vii) Organize the full test, inviting sufficient subjects (assessors), with pre-screening to eliminate those with incompatible attributes.
- (viii) Run the tests (including assessor briefing and dummy runs) and collect the results.
- (ix) Perform post processing to remove outliers and assess significance.
- (x) Produce test report fully documenting laboratory conditions, test conditions and analysis/screening methods employed.

These important stages are discussed in more detail below.

#### 5.07.3.1 Test material

Test material should be selected that is appropriate to the problem or application area being addressed. It will normally be defined in terms of its fixed parameters, which include: the number of sequences used, sequence duration, sequence content, spatial resolution, temporal resolution, and the bit depth.

For most general assessments, the test material will be “critical, but not unduly so” [8]. This means that it should contain some content that is difficult to code while remaining representative of typical viewing conditions. The sequences selected should always include critical material since results for non-critical material cannot usually be extrapolated. In cases where the tests are intended to characterize performance for specific difficult content, then it would be expected that the material used would be selected to reflect these cases.

Generally, at least four types of sequence are selected for testing. This number will provide a minimum coverage of activity levels while not boring the assessors with an over-long test session. Regarding the sequence duration, a length of 10 s is suggested by ITU-R BT.500.

The amount of spatial and temporal activity present in a clip has a high impact on the perceptual quality of the test content. In general, test sequences should be selected that are consistent with the range of channel conditions prevailing in the application areas of interest. It is useful, prior to final selection of test material, to formally assess the spatio-temporal activity levels of the candidate clips to ensure that these cover the appropriate spatio-temporal information space [9, 10].

Following the approach recommended in ITU-T Rec. BT.910 [10], a spatial information (SI) measure is often used, based on the standard deviation of each frame in the sequence after Sobel filtering. The maximum value of the standard deviation, over all frames, is selected as the SI metric. The temporal information (TI) measure used in BT.910 [10] is based on the standard deviation of the difference, over consecutive frames, between co-located luma pixel values.

It is usual to produce an SI vs TI plot to ensure that the sequences selected for testing provide adequate coverage of the SI-TI space. It should be noted that slightly different approaches to that described above

are used by Winkler in [9], based on the mean of the Sobel-filtered frame for SI and the motion vectors for TI. He also includes an information measure based on color.

### 5.07.3.2 Test conditions

The test conditions refer to those parameters that will change during the test session. These would typically include codec types, codec parameters, bit rates, and packet loss ratio during transmission. A typical subjective test would compare two or more codecs or codec configurations for a range of sequences and coding rates.

It is normal to evaluate codec performance over a representative, well distributed set of test conditions. However, most assessment methods are sensitive to the range and distribution of conditions seen. The sensitivity can be reduced by restricting the range of conditions but also by including some explicit (direct anchoring) extreme cases or by distributing these throughout the test without explicit identification (indirect anchoring). Also the larger the number of test conditions, the longer the test session becomes. Tests must avoid viewer fatigue and this imposes quite severe constraints on the number of sequences employed and the number of test conditions evaluated.

### 5.07.3.3 Subject selection

Depending on the nature of the test, observers may be expert or non-expert. Studies have found that systematic differences can occur between different laboratories conducting similar tests [8]. The reasons for this are not fully understood, but it is clear that expert observers will view material differently to non-experts. Other explanations that have been suggested include gender, age, and occupation [8]. In most cases, for consumer applications, it is expected that the majority of observers should be non-expert and that none should have been directly involved in the development of the system under test.

Before final selection of the assessors, all candidates should be screened to ensure that they possess normal visual acuity (with or without corrective lenses). This can be done using a Snellen Chart for visual acuity and an Ishiara Chart to check for color vision. The number of assessors used depends on the scale of the test and the sensitivity and reliability of the methodology adopted. Normally it is recommended that at least 15 subjects are employed [8]. The number can be slightly higher to allow for outlier removal during results processing.

### 5.07.3.4 Testing environment

Testing environments are normally specified according to the requirements of the test—usually meaning either a realistic consumer environment or laboratory conditions. The former provides viewing conditions closer to typical consumer-end environments, while the later is to enable maximum detectability of impairments. Parameters associated with these two environments can be found in [8, 10].

### 5.07.3.5 Testing methodology

In a typical subjective video quality assessment experiment, a video source delivers the presentation clips, either directly to the subject or via a system under test that introduces impairments dependent on the test conditions set at the time. The subject or subjects will view the content on an assessment display

which shows the impaired video clip and, in some cases, also the unimpaired clip, either simultaneously or sequentially. In some cases the user may be able to influence the order, repetition and timing of clips.

Test methodologies broadly fall into two categories—based on their use of single or multiple stimuli. Before we consider these, there are a few general points that are common to all methods.

*Duration of test:* To avoid fatigue, BT.500 recommends that the duration of the test session should be limited to 30 min. In practice, it is not unusual for tests to consume more time, and 40–45 min is not uncommon, especially in cases where the observer is permitted to control the display and revisit sequences multiple times (e.g., the Subjective Assessment Methodology for Video Quality (SAMVIQ) [2]).

*Preparing assessors:* Prior to the test, assessors should receive full instructions on the reason for the test, its content, the types of impairment that will occur and the method of recording results. The first part of any test should include dummy runs which familiarize the assessor(s) with the methodology. These also help to stabilize the observer's opinions.

*Presentation order:* It is normal during the test session to randomize the presentations—within and across tests. This ensures that any influences of fatigue or adaptation are balanced out. It is also normal to include anchors (extreme cases). In some cases the assessor will know which presentations contain anchors, but this is usually not the case.

*Recording opinions:* The opinions of each assessor must be recorded for post-test analysis. This is often done using a line bisection process based on gradings. The observer is asked to place a mark on the line for each presentation assessed and this typically falls into one of five quality scale. The final grade is normally scaled to the range 0–100. In other cases, an automated user interface will be provided, such as that in [2] for the SAMVIQ methodology.

*Recording test conditions:* Because of the inconsistencies that can exist between subjective trials, it is vital to record all salient information about the test and the parameters and equipment used. Good examples of this can be found in Refs. [11, 12].

#### 5.07.3.5.1 Single stimulus methods

In single stimulus methods, there is no explicit anchor and the assessor is presented with a randomized sequence of test conditions, normally including the original. The Absolute Category Rating (ACR) in ITU-T Rec. P.910 does this using 10s clips across a range of test conditions, with voting on the same scale as DSCQS. A Single Stimulus Continuous Quality Evaluation method is proposed as part of ITU-R Rec. BT.500. This is intended to better capture the time and content variations that happen in practical content delivery. Here it is recommended that the test session is organized with longer programme segments (e.g., sport) of around 5 min duration. These are concatenated into a test session of duration 30–60 min comprising a range of programme segments with various quality parameters. One of the main advantages of single stimulus methods is reduced testing time.

SAMVIQ [2] is a form of single stimulus method, but one where the assessor has some control over viewing order and repetitions. The test is organized as a series of test sequences which are assessed in a given order and where the assessor cannot proceed to the next sequence until the previous one has been completely assessed. Within each sequence a number of algorithms and test conditions can be presented in any order as selected by the assessor (however the selection buttons are randomized for each new sequence evaluated).

#### 5.07.3.5.2 Double stimulus methods

Double stimulus evaluations remain the most popular means of evaluating compressed video quality. The most commonly used procedure is ITU-R Rec. BT.500 which, although originally intended for television applications, is now used more widely. ITU-T Rec. P.910 was targeted specifically at multi-media applications, but shares many similarities with BT.500. ITU-R Rec. BT.500 describes two main double stimulus methods: Double Stimulus Continuous Quality Scale (DSCQS) and Double Stimulus Impairment Scale (DSIS).

The DSCQS methodology is best suited for cases where the qualities of the test material and the original are similar and the aim is to assess how well the system under test performs relative to the original. The test is arranged as a sequence of paired clips (A and B) one of which is the original (anchor) and the other is the original impaired by the system under test. The assessor does not know which clip is the original as the order is randomized.

DSIS is similar to DSCQS except that the pair is presented only once and the assessor knows which clip is the anchor as it is always shown before the impaired version. DSIS is generally better suited to assessing the robustness of a system or the effects of more noticeable distortions such as those imparted by transmission errors. This method is very similar to the Degradation Category Rating method (DCR) in ITU-T Rec. P.910.

Pair comparison methods are also double stimulus, but in this case they are based on comparisons between two systems under test for the same test conditions.

#### 5.07.3.5.3 Triple stimulus methods

Some authors have proposed that triple stimulus methods provide increased consistency in results, particularly, for example, when comparing interlaced and progressive scanning methods. The Triple Stimulus Continuous Evaluation Scale methodology (TSCES) was proposed by Hoffman et al. [13], and simultaneously displays the video processed by the system under test alongside (usually they are stacked vertically) two extreme anchors—the original and a low quality coded version. The conventional five grade scale is also used and the assessor is asked to score the system under test in the context of both anchors.

#### 5.07.3.6 Statistical analysis of subjective results

It is essential that the data collected is analyzed and presented in a robust and consistent fashion. Most test results are scaled to the range 0–100 and that is assumed here.

##### 5.07.3.6.1 MOS and DMOS

The mean score across all observers for each presentation is given by:

$$\bar{y}(s, h, r) = \frac{1}{N} \sum_{n=1}^N y(s, h, r, n), \quad (7.1)$$

where  $y(s, h, r, n)$  is the score for observer  $n$  in response to sequence  $s$  under test condition  $h$  for repetition  $r$ , and  $N$  is the total number of observers. This is then similarly processed after screening of observers to produce  $\bar{y}(s, h)$ , the Mean Opinion Score (MOS) for each sequence under a given test condition.

It is common in many tests to compensate scores relative to the reference content. This is particularly important for single stimulus methods where a process of hidden reference removal is applied to produce Difference Mean Opinion Scores (DMOS).

#### 5.07.3.6.2 Confidence interval

When the results of a study are presented, it is good practice to also include the confidence interval [14]. The confidence interval for a mean is the interval that will contain the population mean a specified proportion of the time, typically 95%. Confidence intervals are based on the size of each sample and its standard deviation and provide more information than point estimates. The 95% confidence interval is given by:

$$[\bar{y}(s, h, r) - \delta(s, h, r), \bar{y}(s, h, r) + \delta(s, h, r)], \quad (7.2)$$

where

$$\delta(s, h, r) = 1.96 \frac{\sigma(s, h, r)}{\sqrt{N}}$$

The standard deviation for each presentation is given by:

$$\sigma(s, h, r) = \sqrt{\sum_{n=1}^N \frac{(\bar{y}(s, h, r) - y(s, h, r, n))^2}{(N-1)}}. \quad (7.3)$$

The absolute difference between the experimental mean and the true mean (based on an infinite number of observers) is smaller than the 95% confidence interval of Eq. (7.2) with a probability of 95% (on the condition that the scores are normally distributed).

#### 5.07.3.6.3 Screening of observers

The screening (used in DSIS and DSCQS) for a normally distributed set of scores conventionally uses the  $\beta_2$  test, based on the kurtosis coefficient. When  $\beta_2$  is between 2 and 4, the distribution can be assumed to be normal. The kurtosis coefficient is defined as the ratio of the fourth order moment to the square of the second order moment. Thus:

$$\beta_2 = \frac{m_4}{(m_2)^2}, \quad (7.4)$$

with

$$m_x = \frac{\sum_{n=1}^N (y(s, h, r, n) - \bar{y}(s, h, r))^x}{N}. \quad (7.5)$$

We now compute  $P_n$  and  $Q_n$  as given in Algorithm 1 in order to determine whether any observers should be rejected in forming the distribution. In Algorithm 1,  $N_s$  stand for the number of source sequences,  $N_h$  is the number of test conditions, and  $N_r$  is the number of times each condition is repeated.

---

**Algorithm 1:** Screening of observers in subjective trials using the  $\beta_2$  test [8].

---

```

repeat for each observer, n
    n = n + 1
    for s, h, r = 1, 1, 1 to S, H, R do
        if  $2 \leq \beta_2(s, h, r) \leq 4$  then
            if  $y(s, h, r, n) \geq \bar{y}(s, h, r) + 2\sigma(s, h, r)$  then
                |  $P_n = P_n + 1$ 
            end
            if  $y(s, h, r, n) \leq \bar{y}(s, h, r) - 2\sigma(s, h, r)$  then
                |  $Q_n = Q_n + 1$ 
            end
        else
            if  $y(s, h, r, n) \geq \bar{y}(s, h, r) + \sqrt{20}\sigma(s, h, r)$  then
                |  $P_n = P_n + 1$ 
            end
            if  $y(s, h, r, n) \leq \bar{y}(s, h, r) - \sqrt{20}\sigma(s, h, r)$  then
                |  $Q_n = Q_n + 1$ 
            end
        end
        if  $\frac{P_n+Q_n}{N_s N_h N_r} > 0.5$  and  $\left| \frac{P_n-Q_n}{P_n+Q_n} \right| < 0.3$  then
            | reject observer n
        end
    end
until n = N;

```

---

## 5.07.4 Subjective datasets

A reliable subjective database has three primary uses. Firstly it will produce a robust comparison of the selected compression methods in the context of the test conditions employed. Secondly it provides a very useful basis for validating objective metrics, based on well-established statistical methods. Finally it can be utilized to characterize HVS properties in the context of compression and hence provides a valuable tool for refining objective quality metrics.

Publicly available subjective databases should provide a range of data including reference and distorted videos, the corresponding subjective opinions (in terms of DMOS or MOS) and standard deviations (or standard errors). These are essential for evaluating objective quality metrics. Test environment parameters, such as display screen size and viewing distance also need to be available, as these may be used as parameters for some perception-based quality metrics.

### 5.07.4.1 Databases

Table 7.2 summarizes the characteristics of the primary public databases that are available for general research purposes. A brief overview of these is presented below. The reader is referred to the appropriate reports associated with each trial for further details. An excellent critique and comparison of currently available databases is provided by Winkler in [9].

**Table 7.2** Summary of Primary Subjective Video Databases [9]

<b>Database</b>	<b>Year</b>	<b>Subjects</b>	<b>Scores</b>	<b>SRC</b>	<b>HRC</b>	<b>Resolution</b>	<b>Method</b>
VQEG-FR I [15]	2000	144/153	320	20	16	720 × 576i/720 × 486i@25/30fps	DSCQS
IVC-HD [21]	2008	28	192	24	7	1080i@25fps	ACR-HR
EPFL-PoliMI [22]	2009	34	156	12	12	CIF/4CIF(p)@25/30fps	ACR-HR
LIVE [18]	2010	29	150	10	15	768 × 432p@25/30fps	ACR
VQEG-HD [24]	2010	24	740	49	75	1080i/p@25/30fps	ACR-HR
IVP [23]	2011	35	138	10	10/14	1080p@25fps	ACR

Scores, number of videos with subjective scores; Subjects, number of subjects with valid rating; ACR-HR, absolute Category Rating with hidden reference; SRC, number of source videos, HRC, number of test conditions for each source video.

#### 5.07.4.1.1 VQEG FRTV

The earliest major subjective database for objective video quality assessment was generated via the Video Quality Experts Group (VQEG) FRTV Phase 1 programme [15] (followed by phase 2 in 2003 [16]). The phase I database was constructed in 2000 to address quality issues associated with the introduction of digital TV standards world-wide.

Although the FRTV-I database exhibits good coverage and uniformity [9], it does have limitations in the context of contemporary coding requirements. (i) The artifacts presented do not fully reflect recent advances in video compression as coding is primarily based on MPEG-2. (ii) Its assessments are based on standard definition formats—it does not include high definition material. (iii) Its content is interlaced, whereas the current trend is toward progressive formats, especially for streaming applications.

However despite these limitations, because of the large number of sequences, the large number of subjects, its good coverage and its good uniformity [17], VQEG FRTV Phase I is still one of the more commonly used databases for objective quality metric validation.

#### 5.07.4.1.2 LIVE

A more recent and equally important subjective test database is that developed in the Laboratory for Image and Video Engineering (LIVE) [18,19]. The LIVE database presents impairments based on both MPEG-2 and H.264/AVC compression algorithms and includes results for simulated wired and wireless transmission errors. Compared to the VQEG FRTV-I database, only 38 assessors (29 valid ratings) were employed in the viewing trials leading to some increased variability in MOS scores. Based on the analysis in [9], the LIVE database provides a narrower range of source content (in terms of SI and TI), distortion (in terms of PSNR) and MOS, compared to VQEG FRTV Phase I.

#### 5.07.4.1.3 Others

Other databases that can be used for general metric validation include the IRCCyN/IVC HD database [20,21], the EPFL-PoliMI database [22], the IVP database [23], and the VQEG HDTV database [24]. Compared with the VQEG FRTV Phase I and the LIVE video databases, few objective quality assessment methods have been tested on these databases.

Several databases exist that can be used for specific research purposes and are less useful for validation of video metrics. These include the VQEG Multimedia Phase I database [25] (for multimedia applications) and the NYU subjective dataset I–IV [26–29] (for testing video quality variation based on frame rate and quantization step changes).

### 5.07.4.2 Evaluating metrics using subjective databases

Subjective results are frequently used as a benchmark for establishing a relationship between DMOS (or MOS) scores and a specific objective picture quality metric. The scores produced by the objective video quality metric must be correlated with the viewer scores in a predictable and repeatable fashion. The relationship between predicted and DMOS need not be linear, as subjective testing can exhibit non-linear quality rating compression at the extremes of the test range. The linearity of the relationship is thus not so critical, but rather it is the stability of the relationship and a data set's error-variance that determine predictive usefulness.

### 5.07.4.2.1 Regression analysis using a logistic function

BT.500 [8] describes a method of finding a simple continuous relationship between  $\bar{y}_i$  (the mean score) and the metric based on a logistic function. Firstly the range of mean score values is normalized as follows (after screening):

$$Y_i = \frac{(\bar{y}_i - y_{\min})}{(y_{\max} - y_{\min})}. \quad (7.6)$$

Typical relationships between  $Y$  and a given distortion measure  $X$ , generally exhibit a skew-symmetric sigmoid form. Hence the function  $Y = f(X)$  can be approximated by a logistic function of the form:

$$Y = \frac{1}{1 + e^{(X-a_1)a_2}}, \quad (7.7)$$

where  $a_1$  and  $a_2$  are constants that can be quite simply derived from the experimental data [8].

This has been further extended as a weighted least squares procedure in [15], which is given below,

$$Y_i^w = w_i \left[ \frac{b_1 - b_2}{1 + e^{-\frac{X_i - b_3}{|b_4|}}} + b_2 \right], \quad (7.8)$$

where  $\{Y_1, \dots, Y_N\}$  are the mean subjective opinions,  $\{X_1, \dots, X_N\}$  are the quality metric indices for each test sequence, and  $\{w_1, \dots, w_N\}$  are the reciprocals of the standard errors which are used for weighting the logistic function.  $b_1 - b_4$  are coefficients which are obtained during fitting.

### 5.07.4.2.2 Correlation analysis

Judgements of the performance of a particular objective metric relative to a body of MOS values associated with a specific database, are normally based on certain statistical attributes. These are conventionally related to measures of prediction accuracy, monotonicity, and consistency of its fit. The following measures are commonly used:

The Pearson Linear Correlation Coefficient (LCC) is used as a measure of the accuracy of fit of the metric to the subjective scores. It characterizes how well the metric under test can predict the subjective quality ratings. The general form is defined for a set of  $N$  measurement-prediction pairs  $(\hat{Y}_i, Y_i)$  as:

$$\text{LCC} = \frac{\sum_{i=0}^{N-1} (\hat{Y}_i - \bar{\hat{Y}})(Y_i - \bar{Y})}{\sqrt{\sum_{i=0}^{N-1} (\hat{Y}_i - \bar{\hat{Y}})^2} \sqrt{\sum_{i=0}^{N-1} (Y_i - \bar{Y})^2}}, \quad (7.9)$$

where  $Y_i$  would represent the actual MOS (or DMOS) score and  $\hat{Y}_i$  would represent the predicted MOS (or DMOS) score.  $\bar{Y}$  and  $\bar{\hat{Y}}$  are their mean values.

The degree to which the model's predictions correlate with the relative magnitudes of subjective quality ratings is assessed using a rank order metric. This characterizes the prediction monotonicity, i.e., to what degree the sign of differences across tests correlate between the subjective scores and the metric's prediction of them. Conventionally the Spearman Rank Order Correlation Coefficient (SROCC)

is used for this purpose:

$$\text{SROCC} = \frac{\sum_{i=0}^{N-1} (\hat{Y}_i - \bar{\hat{Y}})(\mathcal{Y}_i - \bar{\mathcal{Y}})}{\sqrt{\sum_{i=0}^{N-1} (\hat{Y}_i - \bar{\hat{Y}})^2} \sqrt{\sum_{i=0}^{N-1} (\mathcal{Y}_i - \bar{\mathcal{Y}})^2}}, \quad (7.10)$$

where  $\hat{Y}_i$  is the rank order of  $\hat{Y}_i$ ,  $\mathcal{Y}_i$  is the rank order of  $Y_i$ .  $\bar{\hat{Y}}$  and  $\bar{\mathcal{Y}}$  are their median values.

The *outlier ratio*, OR, effectively measures prediction consistency—how well the metric predicts the subjective scores over the range of content and impairments. An outlier is normally classed as a predicted data point that is greater than a threshold distance from the corresponding MOS point. Conventionally a threshold of twice the standard error ( $e_{Y_i}$ ) of the MOS values is used. So if the number of data points that satisfy Eq. (7.11) is  $N_{\text{OR}}$  then the outlier ratio is simply given in Eq. (7.12)

$$|\hat{Y}_i - Y_i| > 2e_{Y_i}, \quad (7.11)$$

$$\text{OR} = \frac{N_{\text{OR}}}{N}. \quad (7.12)$$

Finally, the accuracy of objective video quality metrics can be assessed by the *Root Mean Squared Error* (RMSE), which is defined in (7.13)

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (\hat{Y}_i - Y_i)^2}{N}}. \quad (7.13)$$

It should be noted that, among these correlation parameters, LCC, OR, and RMSE highly depend on the fitting model used and on the optimization of the fitting coefficients, while SROCC tests the monotonicity of the objective model and is independent of the fitting process. In practice, SROCC is therefore more often the preferred indicator for assessing quality metric performance.

## 5.07.5 Objective quality metrics

Image and video quality assessments play a crucial role in many aspects of image and video processing, in particular related to video coding and communications. They have three primary uses:

- (i) Algorithm development and benchmarking: We have already seen that subjective evaluations, while very effective in characterizing the strengths and weaknesses of competing coding algorithms, are expensive and time consuming. The existence of reliable and consistent objective metrics provides a much simpler means of comparison. Furthermore they can provide guidance regarding the benefits or otherwise of various algorithmic modifications to a given codec (for example the benefits of adding a loop filter or using multiple reference frames).
- (ii) Rate-Quality Optimization: Quality assessments are increasingly needed in the encoding loop to make instantaneous RQO decisions about which coding modes and parameter settings to use for optimum performance given certain content and rate constraints.

- (iii) Streaming control: In the case of network delivery of video content, it is beneficial for the encoder and transmitter to be aware of the quality of the signal at the receiver after decoding. This enables the encoder to be informed of the prevailing channel conditions and make appropriate decisions in terms of rate- and error-control.

Depending on the availability of reference sequence, objective quality assessment methods are generally classified as either full-reference, reduced-reference, or no-reference.

*Full reference methods* are widely used in applications where the original material is available, such as when assessing image and video coding algorithm performance, or during the coding process when making rate-quality optimization decisions.

*No-reference methods* are only employed where reference content is not available [30], for example when evaluating the influence of a lossy communication system at the receiver. It is extremely difficult to produce “blind” metrics and hence their use is generally restricted to specific operating scenarios and distortion types. They do not generalize well and reduced reference metrics are preferable if possible.

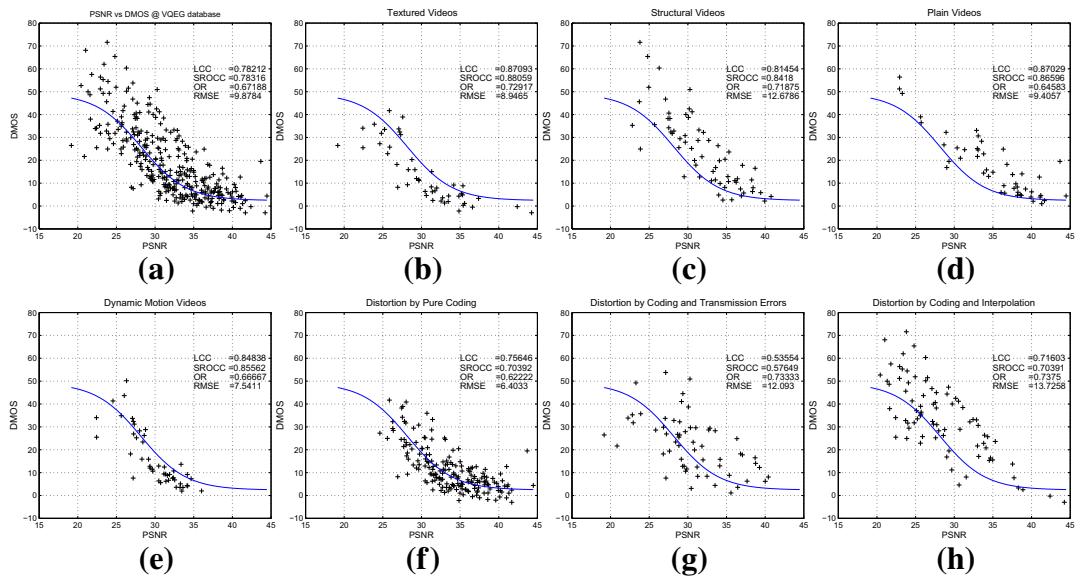
*Reduced reference methods* [31] use partial information about the source during quality assessment. They find application in lossy communication networks where quality predictions can be informed by partial knowledge of the original content and possibly also the channel state. At the decoder, similar features are extracted from the reconstructed signal and compared in the RR metric. An indication of the reconstruction quality at the decoder can then be fed back to the encoder so that it can make informed coding decisions based on the prevailing channel state. Clearly any additional side information places an overhead on the bit rate of the coded information and this must be assessed in the context of the quality gains achieved.

### 5.07.5.1 A characterization of PSNR

For certain types of visual signal under certain test conditions, PSNR can provide a simple and efficient approach to distortion estimation. For example, Huynh-Thu and Ghanbari [32] showed that PSNR can offer consistent results when used to compare between similar codecs or codec enhancements based on the same test data. However, MSE measures can fail badly for certain impairment types, such as a small spatial or temporal shift, an illumination change, or a small variation in a contextual texture [33]. In these cases the perceptual impact may be minor whereas the assessed quality variation could be significant.

Comprehensive overviews of the limitations of MSE-based measures are presented by Girod [34] and Wang and Bovik [33]. Wang and Bovic list the assumptions that underpin the use of MSE: (i) signal quality is independent of temporal or spatial relationships between samples; (ii) signal quality is independent of any relationship between the original signal and the error signal; (iii) signal quality is independent of the signs of the error signal; (iv) all samples contribute equally to signal quality.

Based on results from the VQEG FRTV Phase I database, Zhang and Bull [35] analyzed the correlation between PSNR quality indices and subjective Differential Mean Opinion Scores (DMOS) for a range of conditions. The results are shown in Figure 7.1 where it can be observed that the points fit only loosely and that the fit worsens at lower quality levels (higher DMOS). Zhang and Bull also investigated how content influences PSNR-based predictions by sub-dividing the FRTV-I dataset into five groups based on the dominant content type (spatial texture, structural elements, plain luminance, dynamic motion, or mixed). This grouping was determined subjectively because conventional spatial information (SI)

**FIGURE 7.1**

Scatter plots of subjective DMOS versus PSNR for different groups of source sequences of the VQEG database.

and temporal information (TI) measures do not provide sufficient granularity. The correlations for these content classes are presented in Figure 7.1, where it can be observed that, for videos with significant spatial and dynamic texture, most subjective DMOS scatter points fall below the PSNR predictions. This illustrates the existence of HVS masking effects for static and dynamic textured content. In contrast, for sequences with significant structural content, the PSNR-predicted DMOS values tend to fall below the subjective scores, indicating that (despite edge masking effects) the HVS is very sensitive to errors around structural elements. The scatter plot for plainer content with little high frequency energy, similarly indicates little masking protection.

Zhang and Bull also presented PSNR vs DMOS scatter plots based on different types of coding distortion: compressed content, compressed with transmission errors, and compressed with interpolation. These results are shown in the final three sub-plots in Figure 7.1. PSNR performs reasonably well for pure coding distortion where errors are widely distributed, but provides much poorer predictions for the cases with interpolation errors and transmission failures. It is clear that in the case of highly distorted content, PSNR is not effective and alternative methods are needed.

This analysis confirms that: (i) visual masking exists and is more evident for content with spatial and temporal textures than for plain luminance areas; (ii) highly visible artifacts such as those in plain areas or due to transmission loss tend to cause the HVS to overestimate distortion; (iii) two distinct perceptual strategies are utilized by the HVS—*near-threshold* and *supra-threshold*. It is clear that artifact detection is more important for cases of lower perceptual quality.

### 5.07.5.2 Perception oriented image and video quality metrics

The principle aim of most video quality metrics is to correlate well with visual perception over a wide range of conditions and impairments. Secondary requirements might also include low metric complexity and low latency. These can be important for on-line decision making within a codec or transmission system.

Many perceptually inspired objective quality assessment metrics have been proposed in recent years. These often exploit visual statistics and features in both frequency and spatial domains. We classify these methods into four subgroups: HVS-based methods, visual statistical models, image feature-based approaches, and transform-based methods [36,37]. This classification is based on the primary approach used in each category, and it should be noted that many video quality metrics may incorporate more than one method. Some examples from these subroups are described below. For further details the reader is referred to [1,17,37–39].

#### 5.07.5.2.1 HVS-based quality metrics

HVS characteristics such as contrast sensitivity and visual masking have been exploited, both in coding and in quality assessment. For example, CSF weighting has been employed to reflect the HVS sensitivity across the range of spatial and temporal frequencies, and visual masking—the reduction of perceived distortions at certain luminances and in spatio-temporal textures—has also been exploited. Masking effects are more evident in spatially textured regions, where the HVS can tolerate greater distortions than in smoother areas. A similar phenomenon exists with dynamic textures.

When HVS properties are exploited, it is generally found that the metric provides enhanced correlation with subjective judgements, compared to conventional distortion measures such as MSE. It has been noted that this improvement is more significant when the distorted content is similar to the original [40]. When distortion becomes significant, visual attention is more likely to be attracted by visible artifacts. Under these circumstances, it is often more efficient to characterize the artifacts rather than to purely compute distortions. Based on this concept, quality metrics have been developed which emulate the perception process using a two step strategy: *near-threshold* and *supra-threshold*.

The contrast sensitivity and the near-threshold and supra-threshold properties of the HVS were exploited by Chandler and Hemami in their Visual Signal-to-Noise Ratio (VSNR) still image metric [41]. This emulates the cortical decomposition of the HVS using a wavelet transform. A two-stage approach is then applied to assess the detectability of distortions and determine a final measure of visual SNR. VSNR has been evaluated using the LIVE image database with very good results.

Building on the approach used in VSNR, Larson and Chandler [42] developed the Most Apparent Distortion model (MAD). MAD models both near-threshold distortions and appearance-based distortions. It employs different approaches for high-quality images (near-threshold distortions) and low quality images (supra-threshold distortion). These are combined using a non-linear model to obtain final quality index.

MAD was extended to cope with temporal components in [43] where spatio-temporal slices are processed based on the spatial MAD and the results are then weighted using motion information. The temporal MAD index is combined with spatial MAD (computed from individual frames) to obtain spatio-temporal MAD (ST-MAD) for video. Excellent correlation performance with subjective results is reported based on the LIVE video database.

In the context of perceptual video coding, Zhang and Bull proposed an Artifact-Based Video Metric (AVM) [44] using the DT-CWT as the basis for assessment of both conventionally compressed and synthesized content. AVM correlates well with subjective VQEG scores and has the advantage that it can be easily integrated into a synthesis-based framework because of its high flexibility and low complexity due to extensive parameter reuse.

Inspired by AVM, a Perception-based Video quality Metric (PVM) was recently proposed by Zhang and Bull [35]. PVM simulates the HVS perception processes by adaptively combining noticeable distortion and blurring artifacts (both exploiting the shift-invariance and orientation selectivity properties of the Dual-Tree Complex Wavelet Transform) using an enhanced non-linear model. Noticeable distortion is defined by thresholding absolute differences using spatial and temporal masks which characterize texture masking effects, and this makes a significant contribution to quality assessment when the distorted video is similar to the original. Blurring artifacts, estimated by computing high frequency energy variations and weighted with motion speed, are found to improve the overall metric performance in low quality cases when it is combined with noticeable distortion. Importantly PVM, as with its predecessor, AVM, is intended to be used with synthesized as well as conventionally coded material. Early results indicate that these are the only metrics capable of robust performance in this respect.

Other examples of this class of metric include those reported by Kayargadde et al. [45], Karunasekera and Kingsbury [46], Carnec et al. [47], Zhang et al. [48], and Wei and Ngan [49].

### **5.07.5.2.2 Quality assessment methods based on statistical models**

The integrity of structural information in an image or video is an important cue for visual perception. Wang et al. [50] developed an image quality assessment approach, SSIM (Structural Similarity Image Metric), which estimates the degradation of structural similarity based on the statistical properties of local information between a reference and a distorted image. This is an improved version of the previous Universal Image Quality Index (UIQI) [51] and combines three local similarity measures based on luminance, contrast, and structure.

The advantage of SSIM is that it offers superior performance to PSNR in many cases and that it is relatively simple to implement. As such it is probably at the time of writing, the most commonly used non-MSE metric. It has however been recognized that SSIM suffers from a number of problems, particularly that it is sensitive to relative scalings, translations, and rotations. A complex wavelet-based approach, CW-SSIM has been developed to address these issues [52] as well as an enhanced multiscale version (MS-SSIM) [53]. A further extension to SSIM called V-SSIM which also takes account of temporal information [54] weights the SSIM indices of all frames. This metric has demonstrated improved performance compared to PSNR on the VQEG FRTV Phase I database.

Statistical model-based quality metrics also include contributions from Sheikh and Bovic [55], Lu et al. [56], and Shnayderman et al. [57].

### **5.07.5.2.3 Quality feature-based video metrics**

Pinson and Wolf's VQM [7] is an objective method for video quality assessment that closely predicts subjective quality ratings. It is based on impairment filters that combine measures of blurring, jerkiness, global noise, block distortion, and color distortion. VQM computes seven parameters based on different quality features. These are obtained by filtering the impaired and reference videos to extract the property of interest. Spatio-temporal features are then extracted and a quality parameter is obtained for the feature

by comparing the statistics of the filtered original and impaired video regions. In VQEG tests in 2004, VQM demonstrated superior performance in predicting MOS scores compared to all other algorithms evaluated. It has subsequently been adopted as an ANSI and ITU standard.

Other quality assessment methods in this class are reported by authors including Pessoa et al. [58], Okamoto et al. [59], and Lee and Sim [60].

#### 5.07.5.2.4 Transform-based quality assessment methods

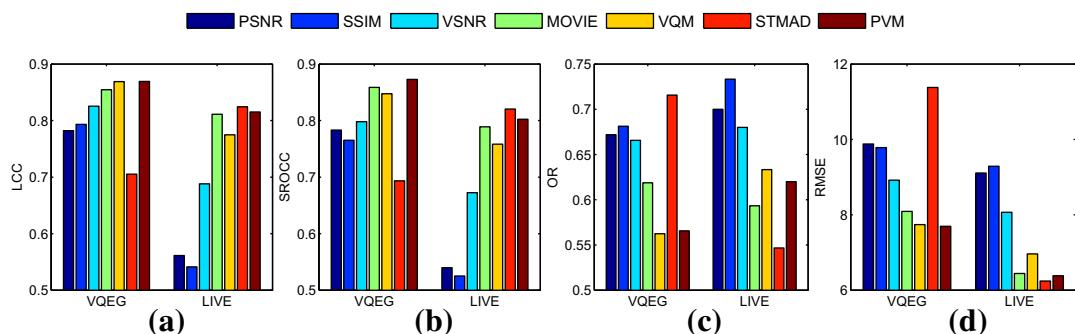
Seshadrinathan and Bovik [19] introduced a motion tuned spatio-temporal quality assessment method (MOVIE). MOVIE analyzes both distorted and reference content using a spatio-temporal Gabor filter family, and the quality index consists of a spatial quality component (inspired by SSIM), and a temporal quality component based on motion information. The accuracy of MOVIE-based predictions has been evaluated using the VQEG FRTV Phase I Database, where it was demonstrated to offer significant correlation improvements compared to both PSNR and SSIM. It also demonstrates excellent performance on the LIVE video database. One issue with MOVIE however is its high computational complexity, due to the large number of Gabor filters used and the temporal extent required for its calculation.

Quality metrics using transform techniques also include MPQM presented by van den Branden Lambrecht and Verschueren [61], DVQ by Watson et al. [62], and the quality model proposed by Lee and Kwon [63].

#### 5.07.5.3 Metric performance comparison

Based on the statistical analysis measures presented in Section 5.07.3.6, seven objective quality assessment methods are evaluated on both VQEG FRTV Phase I and LIVE video databases. These are PSNR, SSIM [50], VSNR [41], MOVIE [19], VQM [7], STMAD [43], and PVM [35]. It is noted that PSNR, SSIM, and VSNR are all image metrics, while MOVIE, VQM, STMAD, and PVM are specifically intended for video quality assessment.

It can be seen from Figure 7.2 that MOVIE, VQM, and PVM perform well on the VQEG database, whereas STMAD, MOVIE, and PVM are the top three performers on the LIVE database. Overall, across both datasets, PVM provides the most consistent performance. STMAD is superior to other metrics on



**FIGURE 7.2**

Objective quality metric performance on both VQEG and LIVE databases in terms of: (a) Linear Correlation Coefficient; (b) Spearman Rank Order Correlation Coefficient; (c) Outlier ratio; (d) Root Mean Squared Error.

LIVE but fails to perform on VQEG. This may be because the material in the VQEG database is interlaced and the parameters in STMAD are tuned according to the LIVE data.

As we discussed at the beginning of this section, objective video quality metrics are not only used for assessing video quality off-line, but can be also utilized in rate-quality optimization for mode selection during the coding process. In this context, compared with MOVIE, VQM, and STMAD, PVM is more compatible with in-loop integration as it only employs a window of five frames to generate one quality index, while the other three methods rely on processing many more frames simultaneously.

For more detail on quality metric performance the reader is referred to [15, 18, 19, 35, 37, 43].

## 5.07.6 Conclusions

This chapter has covered a range of subjective and objective methods for measuring and controlling visual quality. Some of the most common subjective testing methods have been described and it has been shown how these can provide a consistent means of comparing the performance of compression algorithms. The chapter then discussed a number of objective measures that can be used to provide an estimate or prediction of visual quality without the need for costly subjective trials.

MSE-based methods were shown to be useful when used in the context of similar data and coding regimes. However they can also provide deceptive results in many cases, since the perceptual distortion experienced by the human viewer cannot be characterized using such simple mathematical differences. Perception-based video metrics have therefore emerged and a number of these were reviewed and compared, showing improved correlation with subjective scores. However, although perceptual metrics have advanced significantly in recent years, they are still not widely accepted for general picture quality assessment. There are many complex reasons for this. For example, they mostly measure video fidelity (the closeness of the impaired version to the original), they generally measure degradation (i.e., the impaired version is assumed to be worse than the original), and they often do not take account of viewing patterns (viewers tend to focus on regions of interest) or viewing conditions (e.g., lighting).

As we have discussed, one of the most challenging problems for the future is perhaps, to create a reliable, yet low complexity, in-loop quality assessment measure with which to precisely estimate subjective quality and detect coding artifacts. MSE methods are used extensively for this at present, but may become inappropriate if synthesis-based coding becomes more commonplace. For real-time or in-loop processing, quality metrics should be able to: (i) perform assessment at different spatial levels (such as GOP, picture, region or coding unit); (ii) offer manageable computational complexity; (iii) differentiate the effects of an extended video parameter space, including higher dynamic range, frame rate and resolution and take account of different environmental conditions; (iv) ideally provide compatibility with emerging perceptual coding and analysis-synthesis compression methods. These are all topics of ongoing research.

## Additional resources

1. <http://www.its.blrdoc.gov/vqeg/vqeg-home.aspx>. This is the official website of the Video Quality Experts Group. Lots of information on VQEG developed databases.
2. [http://live.ece.utexas.edu/research/quality/live\\_video.html](http://live.ece.utexas.edu/research/quality/live_video.html). This is the homepage of the LIVE Video Quality Database.

## Glossary

Objective quality metric	a measure by which image or video quality can be assessed by way of mathematical models
Rate-distortion optimization	a method of improving quality in image or video compression which minimizes the distortion for a given bitrate at the encoder
Rate-quality optimization	a method of improving quality in image or video compression which optimizes the perceptual quality for a given bitrate at the encoder
Subjective database	a dataset which provides a ground truth of perceptual image or video quality based on extensive subjective ratings. These are often used for the evaluation and benchmarking of objective quality metrics
Subjective quality assessment	a means of evaluating the perceptual quality of images or video via the opinions of human viewers

---

## References

- [1] S. Winkler, *Digital Video Quality: Vision Models and Metrics*, Wiley, 2005.
- [2] F. Kozamernik, P. Sunna, E. Wycken, D. Pettersen, Subjective Quality Assessment of Internet Video Codecs – Phase 2 Evaluations using SAMVIQ, Technical Report, EBU Technical Review, 2005.
- [3] H.L.F. von Helmholtz, *Handbook of Physiological Optics*, first ed., Voss, Hamburg and Leipzig, Germany, 1896.
- [4] H.R. Blackwell, Luminance difference threshold, in: D. Jameson, L.M. Mervich (Eds.), *Handbook of Sensory Physiology*, Springer-Verlag, New York, 1972, pp. 78–101.
- [5] G. Buchsbaum, An analytical derivation of visual nonlinearity, *IEEE Trans. Biomed. Eng.* **BME-27** (5) (1980) 237–242.
- [6] D.H. Kelly, Motion and vision. II. Stabilized spatio-temporal threshold surface, *J. Opt. Soc. Am.* **69** (10) (1979) 1340–1349.
- [7] M.H. Pinson, S. Wolf, A new standardized method for objectively measuring video quality, *IEEE Trans. Broadcast.* **50** (2004) 312–322.
- [8] ITU-R Recommendation BT.500-11, Methodology for the Subjective Assessment of the Quality of Television Pictures, Technical Report, International Telecommunication Union, Geneva, 2002.
- [9] S. Winkler, Analysis of public image and video database for quality assessment, *IEEE J. Sel. Top. Signal Process.* **6** (6) (2012) 1–10.
- [10] ITU-T Rec. P.910, Subjective Video Quality Assessment Methods for Multimedia Applications, Technical Report, ITU-T, 1999.
- [11] F.D. Simone, L. Goldmann, J.-S. Lee, T. Ebrahomi, Towards high efficiency video coding: subjective evaluation of potential coding methodologies, *J. Vis. Commun. Image Represent.* **22** (2011) 734–748.
- [12] P. Hanhart, M. Rerabek, F.D. Simone, T. Ebrahimi, Subjective quality evaluation of the upcoming HEVC video compression standard, in: Proceedings of SPIE 8499, Applications of Digital Image Processing XXXV, 84990V, vol. 8499, 2012.
- [13] H. Hoffmann HDTV – EBU format comparisons at IBC 2006, Technical Review, EBU, 2006.
- [14] D. Lane, Introduction to Statistics, online ed., Rice University. <<http://onlinestatbook.com/>>.

- [15] Video Quality Experts Group, Final Report from the Video Quality Experts Group on the Validation of Objective Quality Metrics for Video Quality Assessment, Technical Report, VQEG, 2000. <[http://www.its.blrdoc.gov/vqeg/projects/frtv\\_phaseI](http://www.its.blrdoc.gov/vqeg/projects/frtv_phaseI)>.
- [16] Video Quality Experts Group, Final VQEG Report on the Validation of Objective Models of Video Quality Assessment, Technical Report, VQEG, 2003. <[http://www.its.blrdoc.gov/vqeg/projects/frtv\\_phaseII](http://www.its.blrdoc.gov/vqeg/projects/frtv_phaseII)>.
- [17] S. Winkler, P. Mohandas, The evolution of video quality measurement: from PSNR to hybrid metrics, *IEEE Trans. Broadcast.* 54 (3) (2008) 660–668.
- [18] K. Seshadrinathan, R. Soundararajan, A.C. Bovik, L.K. Cormack, Study of subjective and objective quality assessment of video, *IEEE Trans. Image Process.* 19 (2010) 335–350.
- [19] K. Seshadrinathan, A.C. Bovik, Motion tuned spatio-temporal quality assessment of natural videos, *IEEE Trans. Image Process.* 19 (2010) 335–350.
- [20] S. Pechard, R. Pepion, P. Le Callet, Suitable methodology in subjective video quality assessment: a resolution dependent paradigm, in: Proceedings of the International Workshop on Image Media Quality and its Applications (IMQA), Kyoto, Japan, 2008.
- [21] S. Péchar, R. Pépin, P. Le Callet, IRCCyn/IVC 1080i Database, 2008. <<http://www.irccyn.ec-nantes.fr/spip.php?article541>>.
- [22] F.D. Simone, EPFL-PoliMI Video Quality Assessment Database, 2009. <<http://vqa.como.polimi.it>>.
- [23] F. Zhang, S. Li, L. Ma, Y.C. Wong, K.N. Ngan, IVP Subjective Quality Video Database, 2009. <<http://ivp.ee.cuhk.edu.hk/research/database/subjective/>>.
- [24] Video Quality Experts Group, Report on the Validation of Video Quality Models for High Definition Video Content, Technical Report, VQEG, 2010. <<http://www.its.blrdoc.gov/vqeg/projects/hdtv/hdtv.aspx>>.
- [25] Video Quality Experts Group, Final Report from the Video Quality Experts Group on the Validation of Objective Models of Multimedia Quality Assessment, Technical Report, VQEG, 2008. <<http://www.its.blrdoc.gov/vqeg/projects/multimedia-phase-i/multimedia-phase-i.aspx>>.
- [26] Y.-F. Ou, T. Liu, Z. Zhao, Z. Ma, Y. Wang, Modeling the impact of frame rate on perceptual quality of video, in: *Proceedings of IEEE International Conference on Image Processing*, 2008, pp. 689–692.
- [27] Y.-F. Ou, Z. Ma, Y. Wang, A novel quality metric for compressed video considering both frame rate and quantization artifacts, in: *International Workshop on Image Processing and Quality Metrics for Consumer (VPQM'08)*, 2009.
- [28] Y.-F. Ou, Y. Zhou, Y. Wang, Perceptual quality of video with frame rate variation: a subjective study, in: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP10')*, 2010, pp. 2446–2449.
- [29] Y.-F. Ou, Y. Xue, Z. Ma, Y. Wang, A perceptual video quality model for mobile platform considering impact of spatial, temporal, and amplitude resolutions, in: *IEEE IVMSP Workshop*, 2011, pp. 117–122.
- [30] P. Marziliano, F. Dufaux, S. Winkler, T. Ebrahimi, A no-reference perceptual blur metric, in: *Proceedings of IEEE International Conference on Image Processing IEEE*, vol. 3, 2002, pp. 57–60.
- [31] R. Soundararajan, A.C. Bovic, Video quality assessment by reduced reference spatio-temporal entropic differencing, *IEEE Trans. Circuits Syst. Video Technol.* 23 (4) (2013) 684–694.
- [32] D. Huynh-Thu, M. Ghanbari, Scope of validity of PSNR in image/video quality assessment, *Electron. Lett.* 44 (13) (2008) 800–801.
- [33] Z. Wang, A. Bovik, Mean squared error: love it or leave it? *IEEE Signal Process. Mag.* (2009) 98–117.
- [34] B. Girod, What's wrong with mean squared error? *Digital Images and Human Vision*, MIT Press, 1998.
- [35] F. Zhang, D. Bull, Quality assessment method for perceptual video compression, in: *Proceedings of the IEEE International Conference on Image Processing*, 2013.
- [36] D.M. Chandler, Seven challenges in image quality assessment: past present, and future research, *ISRN Signal Process.* (2013).
- [37] S. Chikkerur, V. Sundaram, M. Reisslein, L.J. Karam, Objective video quality assessment methods: a classification, review and performance comparison, *IEEE Trans. Broadcast.* 57 (2011) 165–182.

- [38] W. Lin, C.J. Kuo, Perceptual visual quality metrics: a survey, *J. Vis. Commun. Image Represent.* 22 (2011) 297–312.
- [39] H.R. Sheikh, A.C. Bovik, G. de Veciana, An information fidelity criterion for image quality assessment using natural scene statistics, *IEEE Trans. Image Process.* 14 (2005) 2117–2128.
- [40] T.N. Pappas, T.A. Michel, R.O. Hinds, Supra-threshold perceptual image coding, in: Proceeding of the IEEE International Conference on Image Processing, IEEE, 1996, pp. 234–240.
- [41] D. Chandler, S. Hemami, VSNR: a wavelet-based visual signal-to-noise ratio for natural images, *IEEE Trans. Image Process.* 16 (9) (2007) 2284–2298.
- [42] E.C. Larson, D.M. Chandler, Most apparent distortion: full-reference image quality assessment and the role of strategy, *J. Electron. Imaging* 19 (1) (2010) 011006, 1–21.
- [43] P.V. Vu, C.T. Vu, D.M. Chandler, A spatiotemporal most-apparent-distortion model for video quality assessment, in: Proceedings of the IEEE International Conference on Image Processing, IEEE, 2011, pp. 2505–2508.
- [44] F. Zhang, D.R. Bull, A parametric framework for video compression using region-based texture models, *IEEE J. Sel. Top. Signal Process.* 5 (7) (2011) 1378–1392.
- [45] V. Kayargadde, J. Martens, Perceptual characterization of images degraded by blur and noise: experiments, *J. Opt. Soc. Am.* 13 (1996) 1166–1177.
- [46] S.A. Karunasekera, N.G. Kingsbury, A distortion measure for blocking artifacts in images based on human visual sensitivity, *IEEE Trans. Image Process* 4 (6) (1995) 713–724.
- [47] M. Carnec, P. Le Callet, D. Barba, An image quality assessment method based on perception of structural information, in: Proceedings of the IEEE International Conference on Image Processing, vol. 2, 2003, pp. 185–188.
- [48] X. Zhang, W. Lin, P. Xue, Improved estimation for just-noticeable visual distortion, *Signal Process.* 84 (4) (2005) 795–808.
- [49] Z. Wei, K.N. Ngan, Spatio-temporal just noticeable distortion profile from grey scale image/video in dct domain, *IEEE Trans. Circuits Syst. Video Technol.* 19 (3) (2009) 337–346.
- [50] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (2004) 600–612.
- [51] Z. Wang, A.C. Bovic, A universal image quality index, *Signal Process. Lett.* 9 (2002) 81–84.
- [52] Z. Wang, E. Simoncelli, Translation insensitive image similarity in complex wavelet domain, in: EEE International Conference of Acoustics, Speech and Signal Processing, 2005, pp. 573–576.
- [53] Z. Wang, E.P. Simoncelli, A.C. Bovik, Multi-scale structural similarity for image quality assessment, in: Proceedings of the Asilomar Conference on Signals, Systems and Computers, IEEE, vol. 2, 2003, p. 1398.
- [54] Z. Wang, L. Lu, A.C. Bovik, Video quality assessment based on structural distortion measurement, *Signal Process. Image Commun.* 19 (2) (2004) 121–132.
- [55] H.R. Sheikh, A.C. Bovik, Image information and visual quality, *IEEE Trans. Image Process.* 15 (2006) 430–444.
- [56] L. Lu, Z. Wang, A. Bovic, J. Kouloheris, Full-reference video quality assessment considering structural distortion and no-reference quality evaluation of MPEG video, in: Proceedings of the IEEE International Conference on Multimedia & Expo, IEEE, 2002, pp. 61–64.
- [57] A. Shnayderman, A. Gusev, A. Eskicioglu, Multidimensional image quality measure using singular value decomposition, in: Proceedings of the SPIE—International Society Opt. Eng., SPIE, vol. 5294, 2003.
- [58] A. Pessoa, A. Falcao, R. Nishihara, A. Silva, R. Lotufo, Video quality assessment using objective parameters based on image segmentation, *Soc. Motion Picture Television Eng. (SMPTE) J.* 108 (12) (1999) 865–872.
- [59] J. Okamoto, T. Hayashi, A. Takahashi, T. Kurita, Proposal for an objective video quality assessment method that takes temporal and spatial information into consideration, *Electron. Commun. Japan Part I: Commun.* 89 (2006) 97–108.

- [60] S.-O. Lee, D.-G. Sim, New full-reference visual quality assessment based on human visual perception, in: International Conference on Consumer Electronics (ICCE), IEEE, 2008.
- [61] C. van den Branden Lambrecht, O. Verschueren, Perceptual quality measure using a spatio-temporal model of the human visual system, in: Proceedings of the International Society Optical Engineering, vol. 2668, SPIE, 1996, pp. 450–461.
- [62] A. Watson, J. Hu, J. McGowan, Digital video quality metric based on human vision, *J. Electron. Imag.* 10 (1) (2001) 20–29.
- [63] C. Lee, O. Kwon, Objective measurements of video quality using the wavelet transform, *Optical Eng.* 42 (1) (2003) 265–272.

# Multiple Description Coding\*

# 8

**Neeraj Gadgil, Meilin Yang, Mary L. Comer, and Edward J. Delp**

*Video and Image Processing Lab (VIPER), School of Electrical and Computer Engineering,  
Purdue University, West Lafayette, IN, USA*

---

## Nomenclature

3D	three dimensional
3G	the third generation of mobile telecommunications technology
4G	the fourth generation of mobile telecommunications technology
ADPCM	adaptive differential pulse code modulation
AMR-WB	adaptive multirate wideband
AP	access point
AR	autoregressive
ARC	adaptive redundancy control
AVC	advanced video coding
cdf	cumulative distribution function
CR	cognitive radio
DCT	discrete cosine transform
DPCM	differential pulse code modulation
DSQ	delta-sigma quantization
DWT	discrete wavelet transform
ECMDSQ	entropy-constrained multiple description scalar quantizer
EWNC	expanding window network coding
EZW	Embedded Zero-Tree Wavelet
FEC	forward error correction
GOP	group of pictures
i.i.d.	independent and identically distributed
ICMD	iterative coding of multiple descriptions
IP	Internet protocol
ITU-T	international telecommunication union telecommunication standardization sector
JPEG	joint photographic experts group

---

\*This work was partially supported by Cisco Systems and the endowment of the Charles William Harrison Distinguished Professorship at Purdue University.

KLT	Karhunen-Loève transform
LID	linearly independent descriptions
LOT	lapped orthogonal transform
LSP	line spectral pairs
MANETs	mobile ad hoc networks
MB	macroblock
MCTF	motion-compensated temporal filtering
MD-BRDS	multiple description balanced rate-distortion splitting
MD	multiple description
MDC	multiple description coding
MDCOTCQ	multiple description channel-optimized trellis code quantization
MDCT	multiple description correlating transform
MDMC	multiple description motion compensation
MDPAC	multiple description perceptual audio coder
MDSQ	multiple description scalar quantizer
MDSQRA	multiple description spherical quantization with repetition coding of amplitudes
MDSTCQ	multiple description spherical trellis-coded quantization
MDSVC	multiple description scalable video coding
MDTC	multiple description transform coder
MDTCQ	multiple description trellis-coded quantizer
MDVQ	multiple description vector quantizer
ML	maximum likelihood
MMDSQ	modified multiple description scalar quantizer
MP	matching pursuits
MR-DPCM	mutually refining differential pulse code modulation
MSB	most significant bit
MSE	mean-squared error
NC	network coding
NMR	noise-to-mask ratio
P2P	peer-to-peer
PAC	perceptual audio coder
PCM	pulse code modulation
PCT	pairwise correlating transform
PET	priority encoding transmission
pmf	probability mass function
PNC	practical network coding
PSNR	peak signal-to-noise ratio
PVQ	predictive-vector quantizer
QIMM	quantization index modulus modulation
QoE	quality of experience
QoS	quality of service
QP	quantization parameter
RAT	robust audio tool

RDO	rate-distortion optimization
RLNC	random linear network coding
RRD	redundancy rate-distortion
SCD	significant coefficient decomposition
SD	single description
SDVQ	structured dual vector quantizer
SPIHT	set partitioning in hierarchical trees

### 5.08.1 Introduction and history

In the last couple of decades, there has been a dramatic increase in the amount of multimedia traffic over the Internet. The development of 3G/4G and WiFi networks has caused a further increase in the demand for such content delivery over these channels. A statistical study shows that, in 2011, 51% of global Internet traffic was used for video delivery and that number is predicted to increase in the coming years [1]. In 2011, wired devices accounted for the majority of IP traffic (55%). Traffic from wireless devices will exceed traffic from wired devices by 2014. In 2016, wired devices will account for 39% of IP traffic, while WiFi and mobile devices will account for 61% of IP traffic [1]. This increase in multimedia traffic over error-prone wireless channels and among heterogeneous clients has raised some significant challenges for developing efficient coding techniques for this purpose.

A typical signal (audio, image, or video) transmission system consists of a source encoder, a channel encoder, the transmission channel, a channel decoder, and a source decoder as shown in Figure 1 of [2].

Traditionally, the goal of any source coding is to represent a source sequence by the lowest data rate (bits/pixel or bits/second) for a given reconstruction quality. Such signal compression is achieved by removing the redundancies from a source sequence [3,4]. For example, statistical correlations within a video frame are used to reduce the spatial redundancy and statistical correlations among the neighboring video frames can be used to reduce the temporal redundancy. In addition, orthogonal transformations are used to further reduce the redundancy. For example, in image and video compression, the discrete cosine transform (DCT) can concentrate the signal energy in spatial frequency regions and therefore further reduce the redundancy. To ensure the inter-operability between different manufacturers and devices, a series of coding standards have been developed with the growing requirements of applications [4,5].

A defining characteristic of a wireless channel is the variation of the “channel strength” over time and frequency [6]. This can cause packet loss during signal transmission. For example, in real-time applications such as video chat or live streaming, retransmission of lost packets is not feasible. As a result, only a subset of total packets is available at the receiver, which must reconstruct the signal from the available information.

MDC is designed to retain some redundancies in order to combat the uncertainty of packet delivery over a network. In MDC, a single signal source is partitioned into several equally important descriptions so that each description can be decoded independently at an acceptable decoding quality. The decoding quality is improved when more descriptions are received. The encoded descriptions are sent through the same or different channels. When packet loss occurs, the bitstream is still decodable and any subset of the descriptions can reconstruct the original signal with a reduced quality. Thus, when even one of the descriptions is received, the decoder is still able to decode the stream to provide an acceptable quality

without retransmission. This advantage of MDC is very appealing to real-time interactive applications such as video conferencing, for which retransmission is often not appropriate.

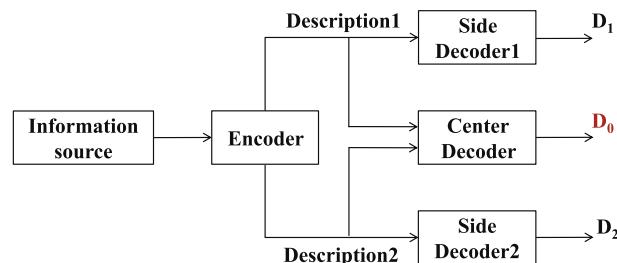
To develop an intuitive notion about multiple descriptions, consider a situation in which the sender desires to convey a message consisting of four symbols ( $S_1 S_2 S_3 S_4$ ) to the receiver. The channel between them suffers from unpredictable losses. So, if the message is sent as one block of symbols, it may not be decodable at all. Now consider that the message is split into two descriptions ( $S_1 S_2$ ) and ( $S_3 S_4$ ) and that these parts are sent independently. In this case, the rate of transmission remains the same. If both descriptions are received, the receiver gets the full message. If one description is received, the receiver gets 50% of the message. Now, to improve robustness, the descriptions are formed by adding some redundancy to each one, giving ( $S_1 S_2 S_3$ ) and ( $S_2 S_3 S_4$ ). If one description is received the receiver gets 75% of the message, but only with the added cost of transmitting one symbol twice. Stretching this argument further, now let the second description be formed as a repetition of the original message. In case of loss of one description, the receiver is able to fully reconstruct the original message. But, as ( $S_1 S_2 S_3 S_4$ ) is sent twice, there's a 100% increase in the rate of transmission. Therefore adding redundancy costs more bits but provides more robustness to the system.

MDC can be posed as a generalization of the Shannon source coding problem [7]. In [8], an achievable rate region is constructed for a memoryless source. A typical structure of MDC is shown in [Figure 8.2](#). The two descriptions are transmitted independently of each other. If one description is received, the side decoder reconstructs the signal with distortion  $D_1$  or  $D_2$ . If both descriptions are received, the central decoder reconstructs the signal with distortion  $D_0$ , where  $D_0 \leq \min\{D_1, D_2\}$ . It is not possible to simultaneously minimize both  $D_0$  and  $(D_1 + D_2)$  [9]. In fact, the performance achieved by using all the descriptions at the decoder can be obtained by the corresponding single description (SD) coder with a smaller total rate [10]. As mentioned earlier, the redundancy introduced in the MDC architecture is meant to improve the error robustness. A detailed theoretical treatment is presented in [Section 5.08.2](#).



**FIGURE 8.1**

A typical media communication system. See [2].



**FIGURE 8.2**

The MDC architecture. See [11].

Like many other communication technologies, MDC was first developed at Bell Laboratories in the late 1970s as an information theory problem for speech communication over the telephone network [10]. Speech coders and information theorists came together to improve the reliability of telephone service without using standby links by splitting the information from a single call and sending it on two separate links. Boyle and Miller designed reliable systems for the physical layer; Gersho and Goodman brought this problem forward to the information theorists [10]. The problem of multiple descriptions was first described at the September 1979 *IEEE Information Theory Workshop* by Gersho, Witsenhausen, Wolf, Wyner, Ziv, and Ozarow [10,12]. Then, with a series of following articles published in *The Bell System Technical Journal*, the first theoretical framework was established. Witsenhausen, in February 1980, proposed the use of two independent channels and analyzed channel breakdown for reliable delivery [13]. Wolf et al. presented a systematic treatment of the problem of multiple descriptions using Shannon's rate-distortion analysis and provided theoretical lower bounds on the distortion [11]. Witsenhausen and Wyner improved the bounds in [14]. Ozarow extended the discussion to Gaussian memoryless sources and presented a complete solution [15]. Jayant applied the concept of multiple descriptions to a speech system with the sample-interpolation procedure that consisted of partitioning of odd-even speech samples [16,17]. In 1982, El Gamal and Cover provided a detailed analysis for achievable rates for multiple descriptions [8,18]. Berger and Zhang analyzed it further in [12,19]. The problem was also studied by Ahlswede in 1985 [20,21].

The second era in the development of MDC started in 1993 with Vaishampayan's design of a multiple description scalar quantizer (MDSQ) [22]. In this approach, two substreams are created by using two indexes, corresponding to two different quantization levels. The quantizer is designed in such a way that if both indexes are received, the reconstruction is equivalent to using a fine quantizer while if only one index is received, the reconstruction is equivalent to using a coarse quantizer [22]. To design the quantizer, a method that uses two quantizers whose decision regions are shifted by half of the quantizer interval with respect to each other was developed by Wang and Zhu [23]. If each quantizer has a data rate of  $R$ , the central distortion is equivalent to that of a single  $R + 1$  bit quantizer and the side distortion is equivalent to that of a single  $R$  bit quantizer. When both the descriptions are received, a total of  $2R$  bits are required to match the performance of a single quantizer with  $R + 1$  bits. Therefore, the coding efficiency is significantly degraded for large values of  $R$ . In this approach, the index streams were assumed to be coded using fixed length codes [23]. To improve the coding efficiency, the MDSQ approach was extended to entropy-constrained multiple description scalar quantizers (ECMDSQs), which use variable length codes for the index streams [24]. The original MDSQ was developed for memoryless sources, with an asymptotic analysis being presented in [25]. For systems with memory, such as communication over a Rayleigh fading channel or a lossy packet network, a multiple description transform coder (MDTC) was proposed by Batllo et al. The coder and its asymptotic analysis is presented in [26,27].

Another popular approach that was developed in 1997 alongside MDSQ exploits a pairwise correlating transform (PCT) to introduce dependencies between two descriptions [28–30]. The broad idea is to divide the transform coefficients into several groups so that the coefficients between different groups are correlated while the coefficients within the same group are uncorrelated. In this way, the lost coefficient groups can be concealed by the received groups and by decorrelating the coefficients within the same group, coding efficiency is maximized. Instead of using the Karhunen-Loëve transform (KLT) that decorrelates all the coefficients, the authors design the transform bases in a way that the coefficients are correlated pairwise. In [28], a pairwise correlating transform is used on each pair of

uncorrelated coefficients generated by KLT. This is realized by rotating every two basis vectors in the KLT with  $45^\circ$ . Each pair of correlated coefficients is then split into two descriptions with equal variance and is coded independently at the same data rate. When both descriptions are received, an inverse PCT is used to reconstruct the coefficients. When only one description is received, the lost description can be concealed based on the correlation between the two descriptions [28]. More general pairwise transforms were proposed in [29] including orthogonal and non-orthogonal approaches. The overhead can be controlled by the number of paired coefficients and the transform parameters. This approach was used for motion-compensated multiple description video coding in [31] by Reibman et al.

Instead of designing the transform basis to introduce correlation among coefficients within the same block, an alternative approach was proposed in order to introduce correlation among the same coefficients in different blocks. Also, to introduce additional correlation, overlapping block transforms can be used. In [32], Hemami et al. proposed the design of lapped orthogonal transform (LOT) bases. In this method, a missing coefficient block is concealed by the average of its available neighbors. A transform is selected based on the channel characteristics, the desired reconstruction performance, and the desired coding efficiency. A LOT-DCT basis proposed in [33] was designed to maximize the coding efficiency, but the complexity is increased by 20–30%.

Along with the development of transforms and coders suitable for MDC, there has also been a significant theoretical interest in developing new bounds and new tools for performance analysis. In 1997, a new theoretical tool, redundancy rate-distortion analysis, was proposed and used to show the performance of PCT. This tool is extensively used in the literature to measure the performance of various MDC systems. A comprehensive review of MDC is presented in [10,23].

With a well-defined problem statement, a well-designed coding framework, and a theoretical toolset to assess the performance, MDC has become a popular topic not only among information theorists but audio, image, video, and network engineers. A plethora of MDC systems has been designed and applied in applications like speech/audio transmission, robust image coding, error resilient video coding, network coding and emerging trends like 3D stereoscopic video and watermarking. In [Section 5.08.2](#), we describe the established theoretical framework of MDC. [Section 5.08.3](#) discusses MDC for speech and audio coding. MDC image coding is described in [Section 5.08.4](#). Applications to error resilient video coding are described in [Section 5.08.5](#). MDC combined with network coding is explained in [Section 5.08.6](#). [Section 5.08.7](#) describes MDC for stereoscopic 3D and [Section 5.08.8](#) summarizes other applications.

## 5.08.2 Theoretical basis

In this section, we describe the information theoretic basis of MDC with the help of rate-distortion analysis. As mentioned earlier, the idea of MDC originated at Bell Laboratories in the late 1970s, where the fundamental framework was proposed by Gersho et al. [8,11,13,15].

### 5.08.2.1 Rate-distortion analysis

As shown in [Figure 8.1](#), source symbols are encoded and transmitted over a channel. The decoder forms a reconstruction signal based on the received encoded data. Shannon's rate-distortion theorem answers

the following question: “What is the minimum transmission rate required so that the reconstructed signal has some specified maximum distortion?”

Let  $X_i, i = 1, 2, \dots$  be a sequence of i.i.d. discrete random variables drawn according to a probability mass function (pmf)  $p(x)$ . Let  $\bar{X}$  represent the vector of  $X$ 's, random variables representing the source symbols and  $\hat{X}$ , the vector of  $\hat{X}$ 's, random variables representing the reconstructed symbols. A distortion measure  $d(\bar{X}, \hat{X})$  is a function of random vectors. Let the rate-distortion function  $R(D)$  be the infimum of all rates  $R$  achieving distortion  $D$ .

**Theorem 1.** *If  $X_i, i = 1, 2, \dots$ , are i.i.d. discrete finite random variables with probability mass function  $p(x)$  then,*

$$R(D) = \inf_{x, \hat{x}} (I(X; \hat{X})), \quad (8.1)$$

where

$$I(X; \hat{X}) = \sum_{x, \hat{x}} p(x, \hat{x}) \log_2 \left( \frac{p(x, \hat{x})}{p(x)p(\hat{x})} \right) \quad (8.2)$$

and the infimum is taken over all joint probability mass functions  $p(x, \hat{x})$  such that

$$E[d(X, \hat{X})] \leq D \quad (8.3)$$

The quantity  $I(X; \hat{X})$ , known as mutual information, determines the achievable rate region for a given distortion [34].  $x$  and  $\hat{x}$  are the source and reconstruction symbols produced with non-zero probabilities.

The problem of MDC can be posed as a generalization of the above theorem. Figure 8.2 describes a typical two description system. Let the encoder transmit two symbol sequences independently at rates  $R_1$  and  $R_2$ . These sequences are such that by observing either one, a decoder can recover an approximation to the source output, but by observing both sequences, the decoder can obtain a better approximation to the source output. Let  $D_1$ ,  $D_2$ , and  $D_0$  be the distortions when the stream of rate  $R_1$ , the stream of rate  $R_2$ , and both streams (with a combined rate of  $R_1 + R_2$ ) are received at the decoder, respectively. Our problem is to determine the set of achievable quintuples  $(R_1, R_2, D_1, D_2, D_0)$ . In other words, “What pairs of rates  $R_1, R_2$  can achieve distortions  $D_1, D_2, D_0$ ?” The rate-distortion region  $\mathbf{R}(\mathbf{D})$  for the distortion set  $\mathbf{D} = (D_1, D_2, D_0)$  is the closure of the set of achievable rate pairs  $\mathbf{R} = (R_1, R_2)$  with each element of the distortion set to be less than the corresponding element from  $\mathbf{D}$ . An achievable rate region is any subset of this rate-distortion region. The following theorem states an achievable rate region for the multiple description problem. Let  $\hat{X}_0, \hat{X}_1, \hat{X}_2$  be three finite reconstruction vectors with the associated distortion measures  $d_m(\bar{X}, \hat{X}_m)$ ,  $m = 0, 1, 2$ .

**Theorem 2.** *Let  $X_1, X_2, \dots$  be a sequence of i.i.d. finite-alphabet random variables drawn according to a probability mass function  $p(x)$ . An achievable rate region for distortion  $D = (D_1, D_2, D_0)$  is given by the convex hull of all  $(R_1, R_2)$  such that:*

$$R_1 > I(X; \hat{X}_1) \quad (8.4)$$

$$R_2 > I(X; \hat{X}_2) \quad (8.5)$$

$$R_1 + R_2 > I(X; \hat{X}_0, \hat{X}_1, \hat{X}_2) + I(\hat{X}_1, \hat{X}_2) \quad (8.6)$$

for some probability mass function  $p(\hat{x}, \hat{x}_0, \hat{x}_1, \hat{x}_2) = p(\hat{x})p(\hat{x}_0, \hat{x}_1, \hat{x}_2|\hat{x})$  and

$$D_1 \geq E[d_1(X; \hat{X}_1)], \quad (8.7)$$

$$D_2 \geq E[d_2(X; \hat{X}_2)], \quad (8.8)$$

$$D_0 \geq E[d_0(X; \hat{X}_0)]. \quad (8.9)$$

The detailed discussion on this with notes on convexification of region and the proof of the theorem is provided in [8]. Note that the distortion  $D_0$  will be smaller than or equal to the single description distortions  $D_1$  and  $D_2$ . If  $D_0$  is chosen to be very close to  $D_1$  and  $D_2$ , the descriptions contain a high amount of correlation (and redundancy) causing  $R_1$  and  $R_2$  to be individually high, making  $R_1 + R_2$  still higher. In this case,  $R_1 + R_2$  is far from the theoretical lower bound to achieve the same  $D_0$  with a single description. This is the case where high redundancy is introduced to improve the robustness at the cost of additional transmission rate. If we allow the single description distortions  $D_1$  and  $D_2$  to be significantly higher than  $D_0$ , the total rate  $R_1 + R_2$  can be brought close to the theoretical lower bound of the single description case. This is equivalent to saying that the descriptions are created such that they are almost uncorrelated with each other producing higher signal compression. So, receiving just one description at the receiver would cause higher  $D_1$  (or  $D_2$ ) because it cannot faithfully reconstruct the message in contrast to receiving both descriptions. This case resembles the no-repetition case of the example described earlier. Therefore, the lower total rate can be achieved by not adding redundancy to the descriptions. This scenario presents little advantage over the conventional single description case in terms of error robustness. It is not possible to simultaneously minimize both  $D_0$  and  $D_1 + D_2$  for a fixed total rate [9]. In fact, the distortion achieved by using all the descriptions at the decoder can be obtained by the corresponding single description coder with a smaller total rate [10]. From this discussion, it becomes clear that controlling the redundancy even theoretically remains one of the main challenges in designing MDC-based systems.

A particular example has been extensively studied in the early literature for checking the tightness of the bounds in [Theorem 2](#) [8, 11, 12, 14, 18, 19]. The specific case has been formulated in [11] as  $R_1 = R_2 = 1/2$ ,  $D_0 = 0$ , and  $D_1 = D_2$ . Thus the source sequence at rate 1 symbol/s is to be encoded into two sequences of rate 1/2 each, such that the original sequence can be recovered error free ( $D_0 = 0$ ) by receiving two sequences. The distortions  $D_1$  and  $D_2$  are to be determined [11]. The distortion is measured by the error frequency criterion [12] i.e., counting the occurrences of error.

$$d_m(X, \hat{X}_m) = 1 - \delta(X, \hat{X}_m), \quad m = 0, 1, 2, \quad (8.10)$$

where  $\delta(X, \hat{X})$  is the delta function, giving non-zero value if  $X$  and  $\hat{X}$  are unequal. Also

$$D_0 = 0, \quad D_1 = D_2. \quad (8.11)$$

If  $D_0 = 0$  were not required, rate-distortion theory [35] would determine  $D_1 = D_2 = D^*$  for  $R_1 = R_2 = 1/2$  with  $D^*$  defined by  $R(D^*) = 0.5$ , where

$$R(D) = 1 - h(D) \quad (8.12)$$

$$= 1 + D \log_2(D) + (1 - D) \log_2(1 - D), \quad (8.13)$$

providing  $D^* = 0.11$  as the solution [12].  $h(D)$  is the entropy corresponding to  $D$  [34]. However, the requirement of error-free recovery necessitates that the information transmitted at 0.5 bits/s over the first channel be independent of that of the second channel [12].

A simple approach would be to make the descriptions consist of alternate source symbols, which allows error-free recovery i.e.,  $D_0 = 0$ . With this scheme, when one description is received the decoder observes alternate symbols and makes an error half the time, producing  $D_1 = D_2 = 1/4$  [11]. It has been proved that  $D_1 = D_2 \geq 1/6$  [11]. After a further analysis, a tighter lower bound of  $D_1 = D_2 \geq 1/5 = 0.2$  has been established in [14]. By applying [Theorem 2](#), an upper bound of  $D_1 = D_2 = (\sqrt{2} - 1)/2 \approx 0.207$  can be shown. The gap between the lower and the upper bound for this distortion is termed the “007 gap” [12, 19]. In [19], the lower bound is proved to be equal to the upper bound i.e., 0.207 eliminating the gap between two bounds. Recall that rate-distortion theory implies that  $R_1 + R_2 \geq R(D_0)$  to achieve distortion  $D_0$ . Now, this example can be thought of as a special case of “no excess rate,” where  $R_1 + R_2 = R(D_0)$ . The “excess rate” situation occurs when  $R_1 + R_2 > R(D_0)$ . For the “no excess rate” case, Berger and Zhang [12, 19] propose a generalized result by deriving an outer bound for the rate region  $\mathbf{R}$ . A portion of the boundary of this outer bound coincides with the curve

$$\left(\frac{1}{2} + D_1 - 2D_0\right) \left(\frac{1}{2} + D_2 - 2D_0\right) = \frac{1}{2}(1 - 2D_0)^2, \quad (8.14)$$

which is a generalization of Witsenhausen’s hyperbola bound for the  $D_0 = 0$  case [12, 13]. The inner bound to  $\mathbf{D}(R_1, R_2, D_0)$  is provided by [Theorem 2](#) [8, 12]. By taking the union of these outer and inner bounds over all  $(R_1, R_2)$  such that  $R_1 + R_2 = R(D_0)$  a projection of  $R$  onto the  $(D_1, D_2)$  plane with a fixed  $D_0$  is determined. This region contains all possible  $(D_1 \geq D_0, D_2 \geq D_0)$  that lie on or above the generalized Witsenhausen’s hyperbola [12]. The specific case on  $D_0 = 0$  has been handled in [20]. Later, it is proved in [21] that the inner bound defined by [Theorem 2](#) is in fact tight for general sources and distortion measures in the no excess rate case. However, for the excess rate case the bound defined in [Theorem 2](#) is not tight and a new inner bound to  $\mathbf{R}$  has been derived in [12]. The following two theorems from [12] support this claim.

**Theorem 3.** *Any quintuple  $(R_1, R_2, D_0, D_1, D_2)$  is achievable if there exist random variables  $\hat{X}_0, \hat{X}_1, \hat{X}_2$  jointly distributed with a generic source random variable  $X$  such that*

$$R_1 + R_2 \geq 2I(X; \hat{X}_0) + I(\hat{X}_1; \hat{X}_2 | \hat{X}_0) \\ + I(X; \hat{X}_1, \hat{X}_2 | \hat{X}_0) \quad (8.15)$$

$$R_1 \geq I(X; \hat{X}_1, \hat{X}_0) \quad (8.16)$$

$$R_2 \geq I(X; \hat{X}_1, \hat{X}_2) \quad (8.17)$$

and there exist  $\phi_1, \phi_2$ , and  $\phi_0$  which satisfy

$$D_1 \geq E[d(X, \phi_1(\hat{X}_0, \hat{X}_1))], \quad (8.18)$$

$$D_2 \geq E[d(X, \phi_2(\hat{X}_0, \hat{X}_2))], \quad (8.19)$$

$$D_0 \geq E[d(X, \phi_0(\hat{X}_0, \hat{X}_1, \hat{X}_2))]. \quad (8.20)$$

**Theorem 4.** For the equiprobable binary source and error frequency distortion measure, in the no excess rate case, all achievable quintuples  $(R_1, R_2, D_0, D_1, D_2)$  satisfy the following conditions:

$$\left(\frac{1}{2} + D_1 - 2D_0\right) \left(\frac{1}{2} + D_2 - 2D_0\right) \geq \frac{1}{2}(1 - 2D_0)^2, \quad (8.21)$$

$$D_1 \geq D(R_1), \quad (8.22)$$

$$D_2 \geq D(R_2), \quad (8.23)$$

where  $D(R)$  is the source's distortion-rate function (inverse rate-distortion function) [12]. The proofs of Theorems 3 and 4 are given in [12].

In case of a Gaussian source with mean-squared distortion, Theorem 2 provides tight bounds [15]. This means that all achievable quintuples  $(R_1, R_2, D_1, D_2, D_0)$  satisfy the conditions stated in the theorem. A further analysis, presented in [36], shows that, as in the single description case, a Gaussian random code achieves the outer bound in the limit as  $D_1, D_2 \rightarrow 0$ ; thus the outer bound is tight in low-distortion conditions.

A generalization of the two-channel results of [8] is discussed in [37], in which an achievable region for the L-channel MDC is presented.

### 5.08.2.2 Redundancy rate-distortion analysis

In an effort to quantify redundancy introduced in a multiple description coder, the redundancy rate-distortion function has been defined and studied extensively [29, 30, 38, 39].

As mentioned previously, a coder minimizing only  $D_0$  using a single description is the standard source coder, and its performance is characterized by the rate-distortion function of the source. Side distortions  $D_1$  and  $D_2$  are significantly higher if this coded sequence is split into two streams and transmitted. Redundancy is added to each stream such that  $D_1$  and  $D_2$  are lowered. Intuitively, the amount of redundancy can be mapped to the added bitrate for the fixed central distortion  $D_0$ . More precisely, the redundancy in coding a source  $Y$  at a central distortion  $D_0$  can be quantified as the difference,  $\rho = R - R^*$ , between the total transmitted bitrate  $R$  and  $R^* = R_Y(D_0)$ , the source rate-distortion function evaluated at  $D_0$ . Note that  $R^*$  is the smallest rate that any coder could use to achieve the distortion  $D_0$ . Without loss of generality, consider the case of  $D_1$ . Let  $\rho(D_1, D_0)$  denote the relationship between redundancy and side distortion  $D_1$ . This function, called the redundancy rate-distortion (RRD) function, describes how many bits of redundancy are required by a coder to achieve a desired  $D_1$  at a given central distortion  $D_0$ . Conversely, the redundancy distortion-rate function describes the achievable side distortion  $D_1$  for a given redundancy  $\rho$  and central distortion  $D_0$  [29].

RRD, as defined in [29], has been used as a tool to assess the performance of MDC coders [30, 38–41]. For the coders presented in this literature, the RRD function is approximately independent of  $D_0$  and can be denoted  $\rho(D_1)$  or  $D_1(\rho)$ . For any fixed  $D_0$  redundancy can range from 0 to  $R(D_0)$ . Redundancy is 0 in the case where the coded stream from an optimal single description coder is split into two substreams. The redundancy is  $R(D_0)$  when the second stream is formed by repeating the one coded using an optimal single description coder. The corresponding side distortion ( $D_1$ ) for  $\rho = 0$  is

$$D_1 = \frac{(\sigma^2 + D_0)}{2} \quad (8.24)$$

and for  $\rho = R(D_0)$

$$D_1 = D_0, \quad (8.25)$$

where  $\sigma^2$  represents the variance of the source variable [30].

RRD also provides a useful framework for understanding the allocation of a total available redundancy to different source variables. A Lagrangian function is used to impose the constraint that for optimal redundancy allocation, each multiple description coder should operate at the same slope on its RRD curve [30].

### 5.08.2.3 Pairwise correlating transforms

MDC requires the presence of some amount of redundancy to provide robustness. Traditional (non-MDC) source coding techniques aim at removing redundancy from the coded bitstream. Standard transform coding methods use a linear transform to decorrelate coefficients to be coded and transmitted. To design suitable MD coders these standard approaches can be modified to force some structured correlation among coefficients instead of completely decorrelating them. Therefore, instead of using the KL transform, an MDC coder can use a transform that produces two sets of coefficients having a specified correlation between sets, while still being uncorrelated within each set [29].

The new transforms, known as pairing transforms, are designed to be used at the output of a KL transform. The KL coefficients are modeled as independent Gaussian variables with different variances. A typical pairing transform design is depicted in Figure 1 of [30].

Consider a matrix  $\mathbf{T}$  as a pairwise MDC transform matrix that takes two independent inputs  $A$  and  $B$  and produces two transformed outputs  $C$  and  $D$

$$\begin{bmatrix} C \\ D \end{bmatrix} = \mathbf{T} \begin{bmatrix} A \\ B \end{bmatrix}. \quad (8.26)$$

The transform  $\mathbf{T}$  controls the correlation between  $C$  and  $D$ , which in turn controls the redundancy of the MDC coder. To perform RRD analysis of pairwise transforms, let  $\sigma_A^2$  and  $\sigma_B^2$  be the variances of zero-mean variables  $A$  and  $B$  with  $\sigma_A^2 > \sigma_B^2$ . In [30], three cases have been considered, namely a general case, optimal transform, and orthogonal transform. In the general case,  $\mathbf{T}$  is parameterized as

$$\mathbf{T} = \begin{bmatrix} r_2 \cos \theta_2 & -r_2 \sin \theta_2 \\ -r_1 \cos \theta_1 & r_1 \sin \theta_1 \end{bmatrix}, \quad (8.27)$$

where  $r_1$  and  $r_2$  are the lengths of the two basis vectors. Let  $\Delta\theta = \theta_1 - \theta_2$ . Let  $\phi$  be the parameter representing the correlation angle between  $C$  and  $D$ . The redundancy per variable is obtained as

$$\rho = \frac{1}{2} \log_2 \frac{\sigma_C \sigma_D}{\sigma_A \sigma_B} = -\frac{1}{2} \log_2 \sin \phi. \quad (8.28)$$

The distortion in the absence of quantization error is

$$D_1 = \frac{r \sin \phi \sigma_B^2}{4 \sin \Delta\theta} \left( \frac{x \sin \phi}{r \sin \Delta\theta} + \frac{r \sin \Delta\theta}{x \sin \phi} \right), \quad (8.29)$$

where  $x = r^2 \cos^2 \theta_1 + \sin^2 \theta_1$ . A complete derivation is presented in [30] with simulation results.

The optimal transform case derives from Eq. (8.29), where an extremum can be achieved by setting

$$x = \frac{r \sin \Delta\theta}{\sin \phi}, \quad (8.30)$$

$$\theta_2 = -\theta_1. \quad (8.31)$$

Thus,

$$D_{1,\text{opt}}(\rho) = \frac{1}{4}((\sigma_A^2 + \sigma_B^2) - (\sigma_A^2 - \sigma_B^2)\sqrt{1 - 2^{-4\rho}}) \quad (8.32)$$

and

$$\mathbf{T} = \begin{bmatrix} \sqrt{\frac{\cot \theta_1}{2}} & \sqrt{\frac{\tan \theta_1}{2}} \\ -\sqrt{\frac{\cot \theta_1}{2}} & \sqrt{\frac{\tan \theta_1}{2}} \end{bmatrix}. \quad (8.33)$$

These transforms are in general not orthogonal. If  $\mathbf{T}$  is an orthogonal transform, then quantization can be performed in the  $C$ - $D$  domain instead of  $A$ - $B$  domain thus reducing the quantization error [30]. Also, the standard transform codes are orthogonal. Therefore, consider an orthogonal transform

$$\mathbf{T} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (8.34)$$

The corresponding side distortion  $D_{1,\text{min,ortho}}$  is derived to be

$$D_{1,\text{min,ortho}}(\rho) = \frac{\sigma_A^2 \sigma_B^2}{\sigma_A^2 + \sigma_B^2}. \quad (8.35)$$

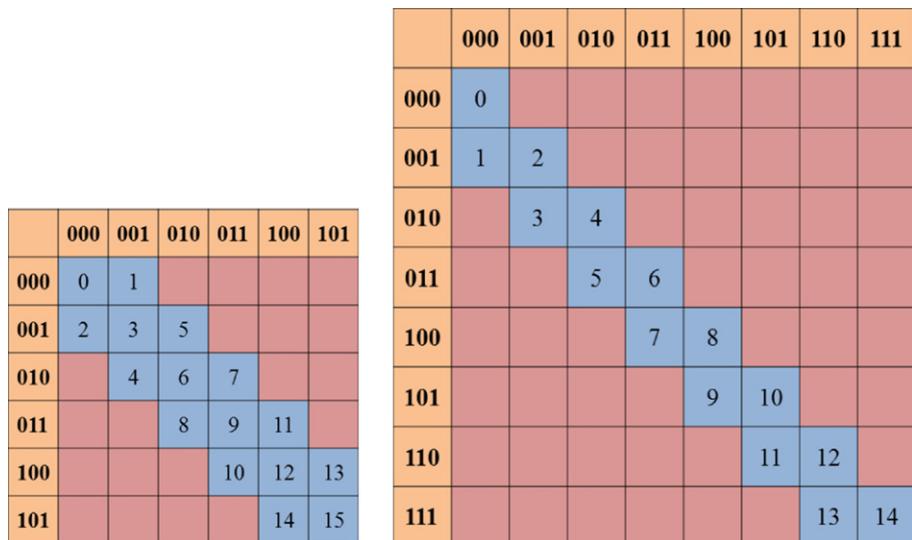
A careful analysis shows that redundancy is used in a balanced, but suboptimal, way to reduce the contributions of both  $\sigma_A^2$  and  $\sigma_B^2$  in side distortion  $D_1$ . For further analysis and a generalization to  $M$  coders, readers should refer to [29, 30, 39].

An extension of this technique to more than two descriptions is presented in [42], in which the authors also present a complete analysis and optimization of the two-description case with an arbitrary source pdf. It also describes the practical use of pairwise correlating transforms (PCT) in image and audio coding.

#### 5.08.2.4 Scalar and vector quantizers

The design of an MD scalar quantizer (MDSQ) is posed as an optimization problem and handled by deriving necessary conditions for optimality. The objective is to design an encoder-decoder pair that minimizes the central distortion  $D_0$ , subject to constraints on the side distortions, ensuring a minimum fidelity. An  $(M_1, M_2)$ -level MDSQ is said to be optimal if it minimizes  $E[d_0]$  subject to  $E[d_1] \leq D_1$  and  $E[d_2] \leq D_2$ , for given values of  $D_1, D_2$ , where  $d_0, d_1$ , and  $d_2$  are random variables expressed as functions of random vectors defined in Section 5.08.2.1. A special case of two descriptions having equal rates is analyzed in [22] using the Lagrangian functional for the constrained optimization problem.

The rate-distortion bound on the average central distortion has been evaluated for a memoryless Gaussian source and squared-error distortion measure [8, 15, 22]. The problem of index assignment can

**FIGURE 8.3**

Examples of two 3-bit quantizers that together give 4-bit resolution. See [10].

be posed as a problem of finding a scanning sequence for a selected set of index pairs that results in a small spread of each cell of each side partition. Consider a set of index pairs constructed from those that lie on the main diagonal and on the  $2k$  diagonals closest to the main diagonal, for some integer  $k$ . With this arrangement, letting  $M_1 = M_2$ , two index assignments are obtained, each of which consists of two types of building blocks. This analysis is presented in detail in [22]. Figure 8.3a depicts an example of MDSQ [10]. This quantization scheme shows the redundancy in the representation through having only 15 of 64 cells occupied. Another example is given in Figure 8.3b that shows an index arrangement with a higher fraction of occupied cells leading to a quantization pair with lower redundancy. Note that, the side distortions in this case are higher than the scheme described in Figure 8.3a because of lower redundancy [10]. The simulation results show that the gap between the optimum MDSQ and the rate-distortion bound is fairly large and improvements can be obtained by designing variable length codes instead of fixed length, or by using vector quantizers.

The first solution of using variable length codes has been presented in [24] which proposes entropy-constrained MDSQ (ECMDSQ), where the outputs of an MDSQ encoder can be further encoded using variable length codes. Necessary conditions for optimality have been derived and an iterative design algorithm to solve for the necessary conditions is presented. The results show a significant improvement over MDSQ [24].

In order to address the complicated index assignment problem, a modified MDSQ (MMDSQ) scheme is proposed in [43]. MMDSQ achieves the same performance as ECMDSQ at high rates and for sources with smooth probability density functions, using a uniform central quantizer. It also offers a more efficient central-side distortion trade-off control mechanism [43].

In [27], the concept of multiple description transform codes (MDTC) is introduced. Consider a vector of source samples  $\mathbf{x}$  such that  $\mathbf{x} = (x(1), x(2), \dots, x(L))^T$  for some  $L$ . The linear transformation  $\mathbf{A}$  leads to the vector  $\mathbf{u} = \mathbf{Ax}$  where  $\mathbf{u} = (u(1), u(2), u(L))^T$ . Each component  $u(l)$  is encoded by the  $l$ th MDSQ encoder  $E_l$ , resulting in a pair of descriptions which are then transmitted over two separate channels [27]. The MDSQ for the  $l$ th transform coefficient operates at a rate of  $R_l$  bits/sample/channel. The rate vector associated with the MDTC is  $\mathbf{R} = (R_1, R_2, \dots, R_l)$ . At the receiver, if one description is lost, the side decoder yields a reconstruction level  $\hat{u}_1(1)$  or  $\hat{u}_2(1)$ . If both descriptions are received, the central decoder gives a reconstructed value  $\hat{u}_0(l)$ . Assuming  $\mathbf{A}$  has been chosen to be invertible, the inverse transformation  $\mathbf{A}^{-1}$  is applied to the vectors of quantized values, providing the vector  $\mathbf{y}_0 = \mathbf{A}^{-1}\hat{\mathbf{u}}_0$  if both descriptions are received,  $\mathbf{y}_1 = \mathbf{A}^{-1}\hat{\mathbf{u}}_1$  if only first description is received, and  $\mathbf{y}_2 = \mathbf{A}^{-1}\hat{\mathbf{u}}_2$  if only second description is received [27].

The asymptotic performance of this MDTC for general sources with memory has been derived. It is shown that the KLT is optimal and the bit allocation is identical to the single channel case. Performance results have been presented for stationary Gaussian sources with memory and for stationary first-order Gauss-Markov sources [27]. In another set of simulations, an asymptotic analysis has been presented for MDSQ for  $r$ th-power distortion measures [44]. For a memoryless Gaussian source and squared-error distortion, there is an 8.69-dB gap between the optimum level-constrained quantizer and the multiple description rate-distortion bound defined in [15]. Under the same set of conditions, a gap of 3.07 dB is reported for the performance of the optimum entropy-constrained quantizer [44].

One solution proposed to improve the performance and close this gap is to construct multiple description vector quantizers (MDVQ). One benefit of a vector quantizer over a scalar quantizer is reduction in granular noise. This is because in higher dimensions it is possible to construct Voronoi cells that are more “spherical” than hypercube [45]. The detailed description of the Voronoi regions of specific lattices is given in [46] and fast quantization algorithms are presented in [47]. For the single description problem, the maximum possible gain over entropy-coded scalar quantization is 1.53 dB [45].

In [45] a general solution is provided to the labeling problem, which plays a crucial role in the design of MDVQ. The labeling problem is to assign a label from the label set to each of the sites. High-rate squared-error distortions for a family of  $L$ -dimensional vector quantizers are analyzed for a memoryless source with probability density function  $p$  and differential entropy  $h(p) < \infty$ . For any  $\alpha \in (0, 1)$  and the rate pair  $(R, R)$ , it is proved that the central distortion  $D_0$  and the side distortion  $D_s$  ( $D_1$  or  $D_2$ ) satisfy:

$$\lim_{R \rightarrow \infty} D_0 2^{2R(1+\alpha)} = \frac{1}{4} G(\Lambda) 2^{2h(p)}, \quad (8.36)$$

$$\lim_{R \rightarrow \infty} D_s 2^{2R(1+\alpha)} = G(S_L) 2^{2h(p)}, \quad (8.37)$$

where  $G(\Lambda)$  is the normalized second moment of a Voronoi cell of the lattice  $\Lambda$  and  $G(S_L)$  is the normalized second moment of a sphere in  $L$  dimensions. An asymptotic analysis reveals that performance arbitrarily close to the multiple description rate distortion bound can be obtained [45]. Design of vector quantizers for asymmetric descriptions is discussed in [48].

A construction of MD trellis-coded quantizers (MDTCQ) is presented in [49, 50]. A tensor product of trellises is used for building a trellis that is applicable to MDC. The Viterbi algorithm [51] provides the best path for encoding. The design procedure utilizes a generalized Lloyd algorithm. The complexity of the scheme is almost independent of the rate [49]. The results are compared with MDSQ [22] and the theoretical bound for a Gaussian source [8]. The plots from [49] indicate that MDTCQ reduces the

gap between MDSQ and the theoretical bound. To further improve the performance of MDTCQ, an MD channel-optimized trellis-coded quantization (MDCOTCQ) has been proposed in [52]. Using the tensor product of trellises, an expression of the multi-channel transition probability matrix is derived. MDCOTCQ is shown to perform much better than MDTCQ in the presence of channel noise [52].

### 5.08.3 Speech coding

As mentioned in [Section 5.08.1](#), MDC was originally developed for improving the performance of the speech telephony system. Jayant [16, 17] proposed an odd-even sample-splitting mechanism for speech coding. Considering the Nyquist sampling theorem [53], the conventional practice in speech telephony is to have a speech signal bandlimited to 3.2 kHz and sampled at 8 kHz. The MDC-based system proposed in [16, 17] has an initial sampling of 12 kHz so that when it is subsampled by 2 preparing two streams of odd and even speech samples of 6 kHz each, it still results in only a slight amount of aliasing. The odd and even samples are compressed using differential pulse code modulation (DPCM) and transmitted independently on two channels. When both descriptions are received at the decoder, DPCM decoding followed by sample interleaving is done. When only one description is received, after DPCM decoding; adaptive linear interpolation is used to predict the lost samples. This is equivalent to using a 6 kHz sampled signal that contains aliasing and quantization noise [10, 16, 17]. This is mathematically formulated in [10]. Consider a discrete-time source sequence

$$x[k] = \alpha_1 x[k-1] + z[k], \quad k \in \mathbb{Z}, \quad (8.38)$$

where  $z[k]$  is a sequence of independent, zero-mean Gaussian random variables. This is a first-order autoregressive (AR) signal model. The scalar constant  $\alpha_1$ ,  $|\alpha_1| < 1$  is the correlation between consecutive samples. The distortion-rate function for this process is [10]:

$$D(R) \geq (1 - \alpha_1^2)2^{-2R}, \quad \text{for } R \geq \log_2(1 + \alpha_1). \quad (8.39)$$

Now, with the separation of odd and even samples,  $x_1[k] = x[2k+1]$  and  $x_2[k] = x[2k]$  are formed. Note that the correlation between two consecutive samples in either description becomes  $\alpha_1^2$  [10]. Letting  $D_0$  denote the central distortion and  $D_s$  denotes side distortion for  $s = 1, 2$ ,

$$D_0(R) \geq (1 - \alpha_1^4)2^{-2R}, \quad \text{for } R \geq \log_2(1 + \alpha_1^2), \quad (8.40)$$

$$D_s(R) \geq \frac{1}{2} \left[ (1 - \alpha_1)^2 + \frac{1}{2}(1 - \alpha_1^2) \right] + \frac{1 + \omega}{2} D_0(R), \quad \text{for } s = 1, 2, \quad (8.41)$$

where  $\omega \in [0, 1]$  [10].

This technique works very well for the range of 2–5 bits/sample (24–60 kbit/s). The quality of half-rate reception in a system designed for total rate  $R$  is similar to that of full-rate reception in a system optimized for total rate  $R/2$ . At 60 kbit/s, the half-rate receiver achieves the desired quality [10, 16, 17].

The design of MDTC is used with packetized speech signals in [25]. First, an  $L$ -dimensional vector is obtained from a continuous-alphabet, discrete-time, stationary ergodic source. Then it is transformed by an  $L \times L$  decorrelating orthonormal transformation matrix. Each transform coefficient is then encoded

using two separate ECMDSQ encoders and the index pair generated by each ECMDSQ is transmitted independently over two channels. The ECMDSQ encoders are designed such that the central distortion  $D_0$  is minimized subject to rate and side distortion constraints. Assuming equal side distortions, i.e.,  $D_1 = D_2$ , an optimum index arrangement is determined for a fixed rate allocation to the transform coefficients. Then, with the assumption of a Gaussian source, an optimum rate allocation is determined [25]. Based on this framework, a forward adaptive MDTC based on the DCT is implemented for 8 kHz sampled real speech. The experimental results show that at  $R = 4$  bits/sample/channel, a gain of 9.15 dB is reported over the DPCM system presented in [16, 17], operating at the same rate and at a packet loss probability equal to  $10^{-3}$  [25].

Multiple description correlating transforms (MDCT) can be employed to improve the concealment performance in case of lost packets. The MDCT allows the correlation and predictability to be continuously adjustable. With this idea, MDCT is employed to improve the existing Bell Labs perceptual audio coder (PAC). A new audio coder multiple description PAC (MDPAC) has been developed. MDPAC achieves considerable perceptual improvement with only a small increase in bitrate when the packet loss probability is moderate [54].

A context adaptive MDC system is proposed in [55], in which each  $2N$ -sample speech segment is split into two components,  $y_1$  consisting of all even samples and  $y_2$  consisting of all odd samples. These two components are first finely quantized e.g., by a PCM or an adaptive DPCM (ADPCM) coder, and packed into two packets. The dequantized data,  $\bar{y}_1$  and  $\bar{y}_2$ , is then used to find the prediction residues,  $r_1(n)$  and  $r_2(n)$  as follows:

$$r_1(n) = y_2(n) - \frac{1}{2}[\bar{y}_1(n) + \bar{y}_1(n+1)], \quad (8.42)$$

$$r_2(n) = y_1(n) - \frac{1}{2}[\bar{y}_2(n-1) + \bar{y}_1(n)]. \quad (8.43)$$

The prediction residues are quantized at a lower bitrate using a coarse quantizer which is a DPCM coder, and is added into the two already created packets. These packets can be decoded independently such that when one packet is received the original  $2N$ -samples can be reconstructed [55]. The results are presented in comparison to the robust audio tool (RAT) and the method proposed by Jayant in [16, 17]. To quantify the reconstruction quality, the noise-to-mask ratio (NMR) has been used. Results show that the proposed scheme achieves the lowest mean NMR when loss rates are lower than 30%. The relative performance gain becomes significant at low rates (smaller than 20%) as compared to higher loss rates, implying that the proposed scheme is more suitable for low loss rate applications [55].

In [56] the authors compare the popular G.729 coding standard, which has its own frame-concealment algorithm, against three different multiple description schemes: repetitive coding, time diversity coding, and residual compensation coding. The results show that when operating at the same bitrate, even the simplest form of MDC provides adequate [56] protection against packet losses.

A novel line spectral pairs (LSP)-based MDC method, which adapts its number of descriptions to network loss conditions in order to conceal packet losses in transmitting low-bit-rate coded speech over lossy packet networks, is proposed in [57, 58]. Based on high correlations observed in linear predictor parameters in the form of LSPs of adjacent frames, multiple descriptions are generated at the sender by interleaving LSPs, and the lost LSPs are reconstructed at the receiver by applying linear interpolation. Without increasing the transmission bandwidth, the proposed scheme represents a trade-off between the quality of received packets and the ability to reconstruct lost packets [57, 58].

To provide reliable end-to-end connections for voice communication over mobile ad hoc networks (MANETs), an MD speech coder based on the adaptive multi-rate wideband (AMR-WB) standard (ITU-T: G.722.2) has been proposed in [59]. This multiple description coder is designed to ensure a full quality at 12.65 kbps if two descriptions are received, and to have a degraded quality at 6.8 kbps if one description is received. A minimal coding redundancy between two descriptions is used. The performance is tested for MANETs with a network simulator and an informal listening test. The results demonstrate that this approach makes effective use of the channel capacity and provides the required reliability [59].

As shown in [Section 5.08.2](#), vector quantization improves the performance of MDC systems over the scalar quantization-based systems. In [60], an MDVQ has been designed to ensure robust audio communication using psycho-acoustically controlled pre- and post-filters that make the mean-squared quantization error perceptually relevant. In the design of the proposed MD coder, the audio signal is first prefiltered and then input into an MDVQ encoder. This coder pairs the samples and outputs two descriptions for every vector created. Then, each description is passed through a lossless coder to remove the redundancy contained in the streams, before transmission over its channel. The lossless coder consists of a predictor and an entropy coder.

A set of detailed experiments can be found in [60]. Comparisons show that with the proposed method, a better rate-distortion operating point is achieved than with a coder with no added redundancy or a coder with the lower half-band repeated over two descriptions or a coder with the lower quarter-band repeated over two descriptions. Since both the psycho-acoustic prefilter and the multiple description scheme add very little delay, an overall delay of the multi-descriptive audio encoder/decoder of 10 ms can be obtained.

A two-channel MD speech coder based on the ITU-T Recommendation G.711 PCM speech coder is proposed in [61]. This coder operates in the PCM code domain in order to exploit the companding gain of PCM. It applies a pair of two-dimensional structured vector quantizers, called structured dual vector quantizers (SDVQ), to each pair of PCM codes, thus exploiting the correlation between adjacent speech samples. If both quantizer outputs are received, they are combined to generate an approximation to the original pair of PCM codes. If only one quantizer output is received, a coarser approximation is still possible [61].

SDVQ-PCM encoding and decoding is illustrated in [61]. There are three main principles behind SDVQ-PCM design. First, incorporating the PCM encoder and decoder, we can reap the benefits of PCM companding. This very basic form of perceptual coding minimizes perceivable quantization noise by distributing small quantization noises to the small speech signal samples and larger quantization noises to the larger samples. The result is a higher perceived speech quality than that provided by uniform quantization at the same data rate. Second, applying a vector quantizer (VQ) to each pair of successive PCM codes ( $c_t, c_{t+1}$ ) (generated from a pair of successive speech samples ( $s_t, s_{t+1}$ )) can exploit the correlation between adjacent speech samples to reduce quantization noise and/or data rate. This helps to combat the data-rate increase that is inherent when replacing one description with two descriptions. Finally, by developing a pair of VQs we can generate a pair of descriptions  $F_1(c_t, c_{t+1})$  and  $F_2(c_t, c_{t+1})$  for each pair of PCM codes ( $c_t, c_{t+1}$ ). Each description  $F_1$  and  $F_2$  carries coarse information about both  $c_t$  and  $c_{t+1}$ . Thus if only  $F_1$  or  $F_2$  is received, a coarse reconstruction of the PCM code-pair is possible. Forcing an appropriate structure on this pair of VQs, can ensure that when both of the coarse descriptions  $F_1$  and  $F_2$  are received, they can be combined to generate a more refined reconstruction of the PCM code-pair [61].

The SDVQ-PCM speech coder offers both options. One could invoke lower-rate coders to accomplish MDC with no increase in data rate. If the complexity of lower-rate coding must be avoided, then the mitigation of longer losses will require some sacrifice in the form of either increased data rate or decreased speech quality. When using 6 bits/sample/channel (for a total data rate of 96 kbps) the coder provides an equivalent PCM speech quality of 7.3 bits/sample when both descriptions are received and 6.4 bits/sample when one description is received [61].

It is shown in [62] that the MDC scheme, together with a new multi-path routing protocol, multi-service business router (MSBR), makes an effective use of the network and provides good performances for real-time voice transmission over a mobile wireless ad hoc network. In [63] an algorithm for designing linear prediction-based two-channel MD predictive-vector quantizers (MD-PVQs) for packet loss channels is presented. This algorithm iteratively improves the encoder partition, the set of multiple description codebooks, and the linear predictor for a given channel loss probability, based on a training set of source data. The effectiveness of the designs obtained with the given algorithm is demonstrated using a waveform coding example involving a Markov source as well as vector quantization of speech line spectral pairs [63]. In another approach [64], phase scrambling is used to decrease the effects of data loss in an audio segment. Phase scrambling allows to spread the data of each block of an audio segment over all other blocks of the scrambled audio. Then it is encoded. The recovered signals are quite satisfactory in terms of qualitative analysis as well as MSE [64].

A new MD scheme based on spherical trellis-coded quantization (MDSTCQ) has been proposed for sinusoidal audio coding in [65] and analytic expressions for the point densities and expected distortion of the quantizers are derived based on a high resolution assumption. The proposed quantizers are of variable dimension, i.e., sinusoids can be quantized jointly for each audio segment whereby a lower distortion is achieved. The quantizers are designed to minimize a perceptual distortion measure subject to an entropy constraint for a given packet loss probability. To overcome the shortcoming of MDSTCQ being suboptimal, another scheme, called spherical quantization with repetition coding of the amplitudes (MDSQRA) is proposed in [66]. Repetition coding is applied to the amplitudes, whereas the phases and the frequencies are coded using MD quantization. MDSQRA outperforms MDSTCQ in terms of measured perceptual distortion for various packet loss probabilities [66].

An interesting application of MDC is found in designing wireless hearing aids. A scheme is proposed in [67] for efficient perceptual coding of audio for robust communication between encoders and wireless hearing aids. To limit the physical size of the hearing aids and to reduce power consumption and thereby increase the lifetime expectancy of the batteries, the hearing aids are constrained to be of low complexity. Therefore, an asymmetric strategy is proposed where most of the computational load is placed at the encoding side. MDC plays a crucial part combating the packet loss problem [67].

Another MD scheme for perceptual coding of audio for real-time robust communication is designed in [68]. This scheme, developed as an alternative to PCM and DPCM and more general noise-shaping converters, involves the use of psycho-acoustically optimized noise-shaping quantizers based on the moving-horizon principle. The moving-horizon construction efficiently incorporates perceptual weighting. In the single description case and without packet losses, it is shown that significant gains over linear PCM could be achieved without introducing delay and without having to change the decoding architecture of existing systems [68]. Hoseok et al. [69] proposes an approach to packet loss concealment for MP3 coded wideband audio by using an MDC based on time domain division with a Wiener filter for frequency compensation. A packet loss protection scheme for interactive audio object rendering

has been proposed in [70]. The approach is used to compress up to five audio objects into two mixture signals to ensure selective reproduction at the reproduction site. The Quality of Experience (QoE) is analyzed using the spectrum distortion of the decoded audio objects in relation to the transmission channel parameters such as packet loss probabilities and the available bandwidth, which is often used for the judgment of Quality of Service (QoS) [70].

#### 5.08.4 Image coding

MDC for image coding was first used with transform coding in [28]. The basic idea is to choose the transform bases so that the coefficients are correlated pairwise. The theoretical treatment of PCT is described in [Section 5.08.2](#) and [30]. The pairwise correlation is achieved by rotating every two basis vectors in the KLT(or DCT) by  $45^\circ$ , followed by splitting of each pair of correlated coefficients into two balanced descriptions. Note that this transformation is such that from any one transformed variable, the original pair can be estimated to a certain accuracy and the total estimation errors from either variable are the same. Also, the transform is unitary so that the mean-squared quantization error added to the transformed variables is the same as that added to the original variables. An optimum transform with these properties is derived as

$$C = \frac{1}{\sqrt{2}}(A + B), \quad (8.44)$$

$$D = \frac{1}{\sqrt{2}}(A - B), \quad (8.45)$$

where  $A$ ,  $B$ ,  $C$ , and  $D$  have the same meanings as described in [Section 5.08.2](#).

The descriptions containing blocks of correlated coefficients are then independently quantized and run-length coded using the respective encoding blocks from the JPEG coder. At the decoder, the inverse of the pairing transform is applied. When only one description is received, the linear predictors are used to conceal the lost paired coefficients. Non-paired lost coefficients are simply set to zero as they do not contribute significantly to the reconstruction [28].

The results are compared with MDSQ from [22] and a DC-coefficient duplicator. They provide evidence that the proposed method achieves a good trade-off between coding efficiency and reconstruction quality in case of only one description received. When compared with a JPEG coder with a lossy channel, the proposed method and MDSQ perform significantly better (more than 12 dB in terms of PSNR) than the JPEG coder. This set of experiments suggests using different rotation angles for the transform basis to provide more flexibility in controlling the correlation, and eventually the trade-off between coding gain versus single channel reconstruction error. Also, with the use of more sophisticated MDSQ, the overhead can be brought down [28].

Two other transform coding methods based on generalized MDC (GMDC) are presented in [71]. The first produces statistically correlated streams so that the lost stream can be estimated from the received data. This method is a generalization of the technique presented in [28] for two channels. The case of coding for four channels is considered. The DCT coefficients separated in both frequency and space are approximately uncorrelated and hence, used to form 4-tuple vectors to which correlating transforms are applied. With a training model using Gaussian source data and numerically optimized transform parameters, the grouping frequencies and the transform for each group are determined.

The second method is based on quantized frame expansions. It considers deterministic redundancy between descriptions as opposed to statistical redundancy used by the previous method. Conceptually, this method is similar to block channel coding, except that the coding is done prior to quantization [71]. While applying this idea to image coding, a frame alternative to a (10, 8) block code is considered so that each length-8 vector is expanded by matrix multiplication to a length-10 vector. Then each length-10 vector is uniformly quantized with a step size depending on the frequency. The results of the experiments are compared with that of a baseline system that uses the same quantization step sizes, but applies (10, 8) block code after the quantization. It is apparent both numerically and visually that the performance of the proposed MD system degrades gracefully as the number of lost packets increases [71].

An image decomposition and reconstruction technique based on lapped orthogonal transforms (LOTs [32]) is developed in [72]. The reconstruction-optimized LOT family provides excellent reconstruction capability, and a transform can be selected based on the loss characteristics of the channel, the desired reconstruction performance, and the desired compression [32]. At the MDC encoder, the bitstream generated by a conventional LOT-based image coder is decomposed so that each description consists of a subsampled set of coded LOT coefficient blocks. Up to four descriptions can be formed comprising of even and odd indexed 2D coefficient blocks. If all the descriptions and hence all the coefficient blocks are available, every  $N \times N$  subblock can be recovered using the direct inverse LOT. In case of lost descriptions, the maximally smooth reconstruction method is applied, which converts the reconstruction problem into an energy minimization problem [72]. This method gives much better results compared to setting lost coefficients to zero. The reconstruction quality from incomplete descriptions depends on the LOT basis used. A detailed analysis of the trade-off between coding efficiency and reconstruction quality in case of various loss rates is presented in [72].

The MDSQ technique described in [Section 5.08.2](#) is used with wavelet-based image coding in [73]. The problem of compressing an image into any number of mutually refinable packets is considered. The goal of this approach is to represent an image in such a way that a faithful reconstruction is possible from an arbitrary subset of packets. First, MDSQ is performed on each subband coefficient creating two descriptions. Then, based on the subband a coefficient belongs to, an appropriate error-correcting code is applied. In an example considered in [73], no code is applied on the high-frequency subband. A simple form of maximum distance separable (MDS) codes,  $(n, 1, n)$  repetition codes, is applied on each description of the low-frequency wavelet subband. Then, the components of each codeword are distributed among multiple arithmetic coders, one per packet to be transmitted. A detailed performance analysis of this MD image coder with other coders is presented in [73].

A bit allocation technique for wavelet-based image coding was developed to find the suitable combination of quantizers in the various subbands that produces the minimum distortion while satisfying the side bitrates, and side distortion constraints [74].

In an approach presented in [75], MDC for images is achieved using pre- and post-processing with an established image coding standard (e.g., JPEG). In this approach, an image is spatially oversampled by zero padding in the two-dimensional DCT domain. Then it is decomposed into two subimages such that odd-row/odd-column pixels and even-row/even-column pixels are assigned to one of the subimages and odd-row/even-column pixels and even-row/odd-column pixels are assigned to the other one. Each of the subimages is then coded into a descriptor using JPEG. The JPEG source decoder does not need any modification. After the decoding of the individual bitstreams, the recombination of the data is

performed. Missing pixel data is estimated from the received pixels [75]. A comparison of the results using this method is presented along with results from a JPEG coder and other MD coders.

The optimal choice of the oversampling factor is determined in [76] so that it minimizes the central as well the side distortions of the source for a given error probability of the channel and the number  $N$  of multiple descriptions. MD image coding using horizontal pixel interleaving and DCT is presented in [77]. Another method combining interleaving and wavelet-based coding is also proposed. In this method, the input image is decomposed to image subbands, and then the low-low subband is fed to a pixel interleaving-based MDC to generate two bitstreams. The other subbands are quantized by the MDSQ and entropy encoded to generate two bitstreams [77]. The partitioning of the coefficients at the encoder is combined with concealment at the decoder in [78]. Based on the statistical characteristics, an optimal concealment method for wavelet coefficients, where the concealment of LL subband coefficients depends on a smoothness measure, is proposed.

A method of creating domain-based descriptions of images is presented in [79]. The descriptions are created by partitioning the transform domain of an image into sets whose points are maximally separated from each other. This partitioning problem is formulated as an optimization problem. A procedure called dispersive packetization for creation of packets from the lattice partition is proposed. The maximal separation property enables simple error concealment at the decoder to estimate the lost descriptions. The experimental results of the proposed method give a better performance, both visually and in terms of PSNR [79].

An unequal loss protected MDC scheme is presented in [80]. In this scheme, two descriptions are formed using X-tree coding [81] and MDSQ. A description is composed of two parts. The first part consists of the critical information about an image i.e., hierarchical tree structures, the most significant bits (MSBs), and signs. The second part is made of the residuals of the significant coefficients. The first part, being critical for reconstruction, is duplicated and sent through both channels, whereas an MDSQ is applied on the second part, forming pairs of quantized indices that are sent over different channels [80]. The PSNR results are compared with significant coefficient decomposition (SCD) presented in [81].

The technique of phase scrambling is employed for MD image coding in [82]. Conceptually, phase scrambling spreads the information in each image pixel among all the pixels of the image. This property is used to create balanced MDs of an image. In [82], phase scrambling is mathematically modeled as a form of all-pass filtering that spreads the information of each pixel across all pixels. The scrambled image is considered to be the circular convolution of the image with a “Key” matrix. Let  $F(i, j)$  represent an  $N \times N$  input image. The Key matrix  $K(i, j)$  is produced by generating a random  $N \times N$  matrix, taking its Fourier transform, setting the magnitude to unity, and taking its inverse Fourier transform. The circular convolution results in a scrambled image matrix  $R$ :

$$R(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} K(m - i, n - j)_C F(i, j), \quad (8.46)$$

where  $K(i, j)_C = K(i \% N, j \% N)$ . Note that each pixel of  $R$  is a weighted sum of all the pixels of the input image, making the probability distribution of the scrambled image a very good approximation of the Gaussian distribution. MDs are formed of the scrambled image and sent across diverse channels. Since the scrambling process does not alter the energy distribution of the image and the correlation of pixels, any energy-compacting, context-based, or entropy coding compression can be applied to the

descriptions. For unscrambling, a similar convolution process is followed by the Fourier transform of the scrambled image and the complex-conjugated Key [82].

The results of the experiments with four descriptions and Embedded Zero-Tree Wavelet (EZW) coder are compared with that of the LOT approach presented in [72] with two, three, and four descriptions; and the JPEG coder. The proposed method outperforms the LOT-based one with a margin of 4–9 dB. It produces images without ringing effects at low frequency, which is attributed to the distribution of localized noise through the scrambling process [82].

An application of MMDSQ mentioned in [Section 5.08.2](#) to image coding is described in [43]. A balanced MDC scheme based on the MMDSQ framework has been proposed in [83]. In this scheme, the constructed descriptions are visually optimized, rather than being optimized with the conventional MSE measure.

### 5.08.5 Video coding

A video signal consists of a sequence of image frames at a fixed frame rate. Successive frames contain spatial and temporal redundancies. A video coder exploits these redundancies to compress the video. A typical encoder contains the corresponding decoder that reconstructs a frame used for prediction of future frames. “Mismatch” is induced in case of packet loss, when the reconstructed frame at the decoder is different than the one at the encoder. The design of an MD video coder raises two main challenges: mismatch control and redundancy allocation. Addressing these issues, a general framework has been developed in [31], using motion-compensated prediction, a prominent feature of most established video coding standards. MDTC-based video MDC, presented in [31], uses three separate prediction paths (and hence, three separate frame buffers) at the encoder, mimicking the three possible scenarios at the decoder. The prediction error is represented in two layers: one generated when both the descriptions are present; and the second when only one description is present. A detailed treatment to this approach is presented in [31, 40]. It also describes three different implementations of this general video coder. Each uses PCT on each pair of DCT coefficients created from an  $8 \times 8$  central prediction error block. The first method uses all three prediction paths to limit the amount of mismatch. The second makes use of generalized PCT introduced in [38]; whereas the third uses only the main prediction loop with some extra redundancy to improve the single reconstruction. The details of these three techniques with simulation results are presented in [31, 40]. The results indicate that it is important to have side prediction loops and transmit some redundancy information about the mismatch. Mismatch control can be advantageous when there is packet loss instead of a complete description loss [31]. When subjected to packet loss, this MD video coder performs significantly better than the traditional SD coders [40].

There have been efforts to summarize MDC approaches and categorize them to simplify and organize the development. In [9], three classes have been defined based on the predictor type.

- Class A aims at achieving mismatch control: The MD splitting is done before motion compensation so that each description can be encoded independently. This method often leads to poor prediction efficiency. In [84], the authors use a simple approach that splits the video sequence into odd and even frames, and separately encodes the two groups to form two independent descriptions. Similar approaches can be found in [85, 86].

- Class B aims at achieving prediction efficiency: The MD splitting is done after motion compensation so that the prediction efficiency is as good as single description coding. But this method requires the reference frames to be fully reconstructed, otherwise mismatch can be induced. In [41], the authors propose the MD-split method, in which motion vectors and low-frequency coefficients are alternated between the descriptions. Another example can be found in [87].
- Class C represents a trade-off between Class A and Class B. In [88], the authors propose a multiple description motion compensation (MDMC) method, where the central predictor forms a linear superposition of the past two reconstructed frames, and the superposition weights control both redundancy and mismatch. Moreover, to reduce mismatch, a compressed version of the mismatch or periodic I-frames can be used.

Another video encoder, proposed in [89], is based on a predictive MD quantizer structure called mutually refining DPCM (MR-DPCM). A detailed background of DPCM-based systems can be found in [25,89] and the analysis of the MR-DPCM system for first-order Gauss-Markov sources with a predefined correlation coefficient is presented in [89]. In the video coder, MR-DPCM is used with the first  $N (< 64)$  coefficients to control the rate overhead (relative to the rate of a single description coder). The remaining  $64 - N$  coefficients are encoded in single description mode. In one packetization strategy, each coded frame is divided into two packets. In another strategy, a slice from an I-frame and the same slice from descendent P-frames are packetized together. The first one offers lower latency as compared to the second, but suffers from loss of synchronization if a packet is lost. The second has better mismatch control. The experiments show that both strategies perform better than a single description coder in the presence of packet loss. The second strategy performs better than the first at the cost of larger reconstruction delay [89].

The coders developed above incorporate spatial and temporal prediction mechanisms into the MD framework. Multiple candidate predictors make this system practically very complicated. The central decoder can use information from both streams to form the best predictor, but at a side decoder, information from the other channel is unavailable and this may cause mismatch [90]. An estimation-theoretic approach to prediction and reconstruction has been presented in [90]. It is advantageous because it takes into account all the information available at each decoder for an optimal estimate, and mitigates the degradation due to quantization in the prediction feedback loop. Then, it is applied to inter-mode video encoding such that the estimation-theoretic predictor is independently implemented for each DCT coefficient causing no virtual loss of optimality. The experimental results show that this method outperforms the conventional prediction methods at both central and side reconstruction in terms of PSNR. Another MDC approach using a second-order predictor is presented in [88]. It predicts the current video frame from two previously coded frames.

The MDC method described in [31] yields a poor prediction when both descriptions are available [91]. An MDC scheme based on a matching pursuits (MP) video coding framework is presented in [91]. Its performance is enhanced using a technique based on maximum likelihood (ML) estimate. The rate-distortion performance is improved for all three cases when both descriptions are available. ML estimate technique works best for low-motion sequences. Two-state Markov and Rayleigh fading channel models are used for simulations. The results indicate that this MDC outperforms SDC in bursty slowly varying environments [91].

In [41], an MD video coder is proposed based on rate-distortion splitting, in which the output of a standard video coder is split into two correlated streams. The problem of formation of unbalanced descriptions due to alternation of non-zero DCT coefficients has been addressed in [92]. In the proposed scheme, known as MD-balanced rate-distortion splitting (MD-BRDS), two descriptions with virtually identical rates and distortions are formed using optimization in the RRD sense. The optimum allocation of redundancy among the blocks in a frame is achieved using Lagrangian relaxation. A greedy algorithm is used to meet the equal distortions criterion. This design complies with the existing DCT-based standards.

The methods described above can provide good performance but they generate descriptions that are incompatible with established video standards such as H.264 [93]. This leads to the use of MDC at pre- and/or post-processing stages [94]. The general idea is to split the video source into two subsequences, which are encoded independently. At the decoder side, when the two descriptions are received, the decoded subsequences are postprocessed to recover the full quality video. When one description is lost, the received one can be used to reconstruct the video at a coarser quality [2]. In [75], an oversampling method is proposed to add redundancy to an image. Then, with a partitioning scheme of the oversampled image, multiple subimages of equal pixel dimensions are created. This has been extended to video applications in [95], which uses zero padding in the two-dimensional DCT domain. In [76], an arbitrary number of descriptions, based on zero padding in the DCT domain followed by MD generation using polyphase downsampling, is described [2]. A simple way to generate two descriptions is to use horizontal or vertical downsampling. To generate more than two descriptions, the partition is done in such a way that redundancy is uniformly distributed along the image columns and rows. Similar ideas can be found in [85, 96]. To protect a region of interest, a content-based multiple description image coder is proposed in [97]. A nonlinear geometrical transform is used to add redundancy mainly to the region of interest, followed by splitting the transformed image into subimages which are coded and transmitted separately [2]. Other downsampling-based MDC in the spatial, temporal, or frequency domain can be found in [79, 84, 98–102].

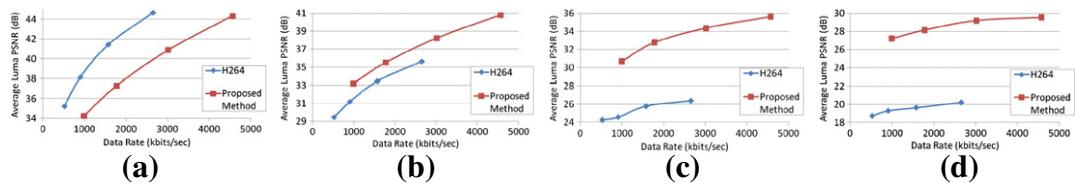
Another classification of MDC methods is proposed in [103] to study various MDC approaches based on the partitioning stage. The video source can be partitioned into multiple descriptions at different stages during encoding. These stages are mainly preprocessing, encoding, and postprocessing. MDC methods can be categorized according to the stage at which the MD partitioning is performed. For MDC methods with descriptions generated in preprocessing, the original video is split into multiple subsequences before encoding and these subsequences are encoded independently to generate multiple descriptions. Typical examples are downsampling-based MDC in spatial and temporal domains as described before. For MDC methods with descriptions generated during encoding, the one-to-multiple mapping is realized by particular coding techniques such as MDSQ, PCT-based MDC, and LOT-DCT-based MDC as have been discussed. For MDC methods with descriptions generated during postprocessing, the one-to-multiple mapping is realized by splitting an encoded bitstream into multiple substreams in the compressed domain. A FEC-based MDC is proposed in [104]. In this method, maximum distance separable ( $n, k$ ) erasure channel codes are used to generate multiple substreams using a joint source-channel coding method. The redundancy can be controlled by  $k$ . These three classes have their own advantages and disadvantages. The main advantage of preprocessing-stage MDC methods is that the coding stage is unaffected. Thus the coded descriptions can be made compatible with existing video standards, which is convenient to implement. MDC methods based on a preprocessing scheme take advantage of the assumption that spatially or temporally adjacent video data samples are highly correlated. Therefore,

video data samples in one description can often be well estimated from the corresponding data samples in other descriptions. One simple way to implement the preprocessing data partition is to split odd and even frames of a video sequence into two subsequences. Each of the newly generated subsequences can be encoded using a video coding standard such as H.264. Some standard-compliant temporal-domain-downsampling MDC methods have been proposed [84, 99]. Another simple way to implement the preprocessing data partition is to split odd and even columns of each frame in a video sequence into two subsequences. Similarly, each of the newly generated subsequences is encoded using a video coding standard. An example of the spatial downsampling-based MDC methods can be found in [105]. Since the correlations among neighboring pixels are reduced after downsampling, coding efficiency is degraded. MDC algorithms based on frequency-domain downsampling can alleviate this correlation-reduction problem. One approach to implement the preprocessing data partitioning in the frequency domain is to add zeros to the transform coefficients in both dimensions after DCT, followed by MD generation using polyphase downsampling [95]. Another approach is to split the wavelet coefficients into maximally separated sets [79] and use simple error concealment methods to produce estimates of lost signal samples.

To combine the advantages of scalable coding and MDC, a multiple description scalable video coding (MDSVC) scheme based on motion-compensated temporal filtering (MCTF) [106] is described in [100]. Video frames are filtered into low-frequency and high-frequency frames, then the high-frequency frames are separated into two descriptions. Lost frames can be estimated from low-frequency frames along with the motion vector information. In [101], the authors describe an MDC method based on fully scalable wavelet video coding, in which post encoding is done to adapt the number of descriptions, the redundancy level, and the target data rate.

Due to the demand for applications in scalable, multicast, and P2P environments, there has been an increasing interest in MDC methods with more than two descriptions [2]. Zhu and Liu [103] propose a multi-description video coding based on hierarchical B pictures where temporal-level-based key pictures are selected in a staggered way among different descriptions. In [103], the authors use a linear combination of received descriptions to optimize decoding results. However, this approach suffers from high data-rate redundancy by duplicating the original sequence into two descriptions. Hsiao and Tsai [107] present a four-description MDC which takes advantage of residual-pixel correlation in the spatial domain and coefficient correlation in the frequency domain.

A four-description MDC framework is developed in [108]. This MDC architecture is a Class A method, having good mismatch control at the expense of coding efficiency. This framework has been extensively used to develop various error concealment techniques for MDC video [108–113]. In this approach, the original video sequence is first temporally subsampled to form two (Odd and Even) subsequences, which are then spatially subsampled by choosing alternate columns (denoted as 1 and 2) of each frame. These four subsequences are encoded independently using an established standard such as H.264 and sent through the same or different channels [108]. When the four descriptions are correctly received, each description can be decoded independently. The final reconstructed video is the combination of the four decoded subsequences. However, when packet loss occurs during transmission, joint decoding is performed. In this case, spatial concealment is performed by applying a two-neighbor bilinear filter and temporal concealment is performed using a non-motion-compensated method, which copies the value at the spatially collocated pixel in a neighboring frame [2, 108]. To assess its packet loss performance, the channel is subjected to simulated packet loss according to Gilbert's model [2, 108, 114].

**FIGURE 8.4**

Packet loss performances comparison for the “Soccer” sequence. (a) Packet loss = 0%, (b) packet loss = 10%, (c) packet loss = 20%, (d) packet loss = 30%. See [2].

**Table 8.1** Adaptive Concealment Scheme

Concealment Methods	Odd <sub>1</sub> + Odd <sub>2</sub>	Odd <sub>1</sub>	Odd <sub>2</sub>	Loss
Even <sub>1</sub> + Even <sub>2</sub>	N/A	Adaptive	Adaptive	Temporal
Even <sub>1</sub>	Adaptive	Spatial	Adaptive	Spatial-Temporal
Even <sub>2</sub>	Adaptive	Adaptive	Spatial	Spatial-Temporal
Loss	Temporal	Spatial-Temporal	Spatial-Temporal	Frame Repeat

In the presence of packet loss, H.264 degrades severely compared with the proposed method. Its packet loss performance is obviously worse when the packet loss rate reaches 10% as shown in Figure 8.4. The scalability performance is evaluated when some of the four descriptions are totally lost, and the others are correctly received. At the decoder, the client can as well decide how many descriptions it wants to receive according to its bandwidth.

The experimental results presented in [2, 108] clearly indicate that temporal concealment works better for low-motion sequences, and spatial concealment works better for high-motion sequences. This is intuitive from the fact that the temporal concealment involves pixel copy (non-motion-compensated method). To be able to choose the right concealment approach when both concealment data are present, several adaptive schemes are proposed in [111–113]. Table 8.1 summarizes the adaptive concealment scheme. In a method based on error tracking [111], the initial concealment distortion is calculated for an error-free reference frame. Let the initial concealment distortion of frame  $k$ , denoted as  $D_k^c$ , be the distortion that would result between the original frame  $k$  and the concealed frame  $k$  when the reference frame for concealment is error free. Here, squared error is used to represent the distortion. For a sequence with the resolution of  $m \times n$ , the initial concealment distortion for frame  $k$  is:

$$D_k^c = \frac{1}{mn} \times \sum_{y=0}^{m-1} \sum_{x=0}^{n-1} (\hat{f}_k(x, y) - f'_k(x, y))^2, \quad (8.47)$$

where  $\hat{f}_k(x, y)$  is the reconstructed pixel of frame  $k$  at the encoder, and  $f'_k(x, y)$  is the concealed pixel of frame  $k$  when the frame used for concealment is error free. For each MDC representation,  $D_k^c$  is obtained at the encoder for each of temporal concealment and spatial concealment for every packet. This information is transmitted along with the encoded bitstream to help the decoder choose the optimal

concealment method. The pixel  $(x, y)$  of P-frame  $k$  is predicted from pixel  $(\tilde{x}, \tilde{y})$  of frame  $k - 1$ . For a received I-frame, as all the predictions are within the frame, there is no error propagation. Since the decoder can decode the frame exactly, the pixel error for frame  $k$  is zero. For P-frame, it is assumed that the reconstructed pixel of frame  $k$  at the decoder is  $\tilde{f}_k(x, y)$  and the received corresponding residue is  $\hat{d}_k(x, y)$ , then:

$$\tilde{f}_k(x, y) = \tilde{f}_{k-1}(\tilde{x}, \tilde{y}) + \hat{d}_k(x, y). \quad (8.48)$$

Therefore the error is:

$$e_k(x, y) = e_{k-1}(\tilde{x}, \tilde{y}). \quad (8.49)$$

If a frame is lost at the decoder, error concealment is used.  $g(x, y)$  is assumed to be the pixel value at  $(x, y)$  from the frame used for concealment,  $\hat{g}(x, y)$  is its reconstructed form at the encoder, and  $\tilde{g}(x, y)$  is its reconstructed form at the decoder. For temporal concealment, the concealed pixel of frame  $k$ ,  $\tilde{f}'_k(x, y)$  equals  $\tilde{g}_k(x, y)$ . Therefore

$$e_k(x, y) = e_k^c(x, y) + e_k^g(x, y). \quad (8.50)$$

Here  $e_k^c(x, y)$  denotes the pixel error from initial concealment, and  $e_k^g(x, y)$  denotes the pixel error of the frame used for concealment. For spatial concealment,

$$e_k(x, y) = e_k^c(x, y) + (e_k^g(x - 1, y) + e_k^g(x + 1, y)) / 2. \quad (8.51)$$

Summing over all pixels in the frame:

$$D_k = D_k^c + D_k^g, \quad (8.52)$$

where  $D_k$  is the distortion of frame  $k$  to be decoded or concealed,  $D_k^c$  is its initial error concealment distortion, and  $D_k^g$  is the distortion due to the frame used for concealment. Each encoded packet is assumed to be lost at the encoder. Temporal concealment and spatial concealment are used separately and each corresponding initial error concealment distortion  $D_k^c$  is recorded. This information is transmitted along with the encoded bitstream to the decoder. After each received frame is decoded, its distortion  $D_k$  is updated. When packet loss occurs, the distortion of choosing each concealment method is the summation of its corresponding initial concealment distortion and the distortion of the frame used for concealment as illustrated in Eq. (8.52). The concealment method with the smaller distortion is chosen, and the corresponding distortion  $D_k$  is updated.

As shown in Figure 8.5 the image produced by the adaptive method is much sharper than the non-adaptive method and the overall visual quality is much better. For a detailed derivation of the equations and analysis of experimental results, the readers should refer to [2, 111].

A frame may contain high motion and low motion at the same time. Therefore, it is advantageous to choose the error concealment method on the macroblock (MB) level. To facilitate this, each macroblock is labeled with a spatial concealment flag ( $S$ ) or a temporal concealment flag ( $T$ ) before encoding. The concealment flags are transmitted as side information to the decoder. In the foreground-background mapping method [112], global motion is first detected. Let  $f_n(i, j)$  be the pixel value at position  $(i, j)$  in frame  $n$ . The index  $i, j$  should be within a macroblock, i.e.,  $16k \leq i, j \leq 16k + 15$  where  $k$  is non-negative integer. If the mean-squared error between two neighboring frames within a macroblock  $\frac{1}{16^2} \sum_i \sum_j (f_n(i, j) - f_{n-1}(i, j))^2$  is greater than threshold  $T_1$ , such a macroblock is assumed to have

**FIGURE 8.5**

Packet loss performance comparison for the “*Bridge*” sequence with identical packet loss positions. The left frame is produced using the proposed adaptive method and the right frame is produced using non-adaptive method. See [111].

significant motion. If the percentage of the macroblocks with significant motion is greater than threshold  $T_2$  as illustrated in Eq. (8.53), this frame is assumed to have global motion.

$$p \left( \left( \frac{1}{16^2} \sum_i \sum_j (f_n(i, j) - f_{n-1}(i, j))^2 \right) > T_1 \right) > T_2. \quad (8.53)$$

For a frame with global motion, spatial concealment is a better choice than temporal concealment. Thus for this type of frame, spatial concealment is used and all the macroblocks within the frame are labeled as “S.” If no global motion is detected, macroblocks with significant motion are classified as foreground and the others as background. Spatial concealment is used for foreground macroblocks and temporal concealment for background macroblocks, as illustrated in Eq. (8.54).

$$\text{flag} = \begin{cases} S & \left( \frac{1}{16^2} \sum_i \sum_j (f_n(i, j) - f_{n-1}(i, j))^2 \right) > T_1 \\ T & \text{others} \end{cases}. \quad (8.54)$$

If  $T_1$  is too small, many background macroblocks are classified as foreground, which could blur the image. If  $T_1$  is too large, many foreground macroblocks are classified as background, which could induce severe mismatch artifact [112]. To approach the optimal threshold, the macroblocks are ranked in a descending order based on the histogram of the mean-squared error and take the top part of macroblocks as the foreground with a certain percentage which is sequence-dependent in the experiments. Both spatial and temporal concealment are done on each uncorrupted macroblock (not corrupted) before encoding and the method with smallest distortion is used as the concealment method [112]. Both the subjective quality and objective quality assessment demonstrate that these two proposed methods greatly improve the non-adaptive method [2, 112]. But these two methods require the transmission of side information to the decoder for better error concealment. Even though the additional data rate is negligible compared to the total data rate, correct delivery of the side information is very important for these two methods.

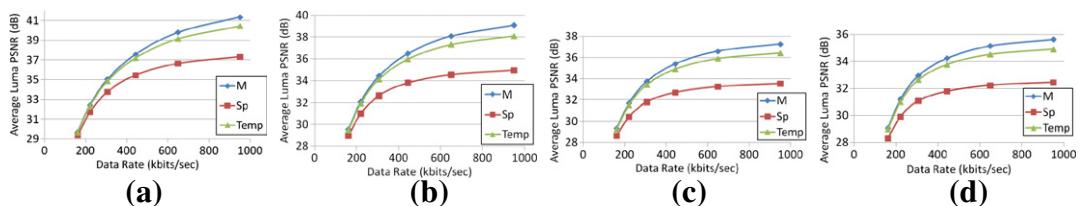
An adaptive method based on motion vector analysis is proposed in [113], which does not require transmission of any side information. In this method, for a received macroblock, the average motion

within it is estimated as the weighted average of all motion vectors associated with different partitions of that MB [113]. Let  $M$  be a macroblock from a received slice and  $N$  be the total number of partitions in  $M$ . Let the  $i$ th partition of  $M$  have pixel dimensions of  $(P \times Q)^i$  and motion vectors as  $((mv_x)^i, (mv_y)^i)$ . Now,  $(\beta_x)_M$ , the average absolute motion in the x-direction for  $M$ , is given by:

$$(\beta_x)_M = \frac{\sum_{i=1}^N |(mv_x)^i| * (P \times Q)^i}{\sum_{i=1}^N (P \times Q)^i}. \quad (8.55)$$

A similar equation applies to the motion in the y-direction  $((\beta_y)_M)$ . For the bi-predictive mode present in a partition, the effective motion vectors are obtained by the weighted average of two motion vectors and used as  $((mv_x)^i, (mv_y)^i)$  in the above equations. The average motion ( $\gamma$ ) within the macroblock  $M$  is the magnitude of the average motion vector  $((\beta_x)_M, (\beta_y)_M)$ . Now, the average motion of a lost MB is estimated as the average of the average motions of the received MBs; that carry the same MB index, but belong to other descriptions. When the concealment data is available from both the concealment methods above, then the lost MB is adaptively concealed based on the amount of motion present in it [113]. If the estimated motion of the lost MB is greater than a globally determined threshold ( $T$ ), it is concealed using spatial concealment. If the estimated motion is less than  $T$ , temporal concealment is used. [Figure 8.6](#) shows the packet loss performance at packet loss rate from 5% to 20% for the “News” sequence. For the “News” sequence, default temporal concealment (*Temp*) clearly outperforms default spatial concealment (*Sp*), but the proposed method (denoted by *M*) does better than *Sp* and *Temp*. This is indicative of the fact that this method is adaptive to the amount of motion present in the sequence. For a detailed analysis and results, the readers should refer to [113].

A trade-off between error resilience (robustness) and coding efficiency is a main challenge for MDC. Many recent MDC algorithms take into account adaptive redundancy control (ARC) to optimize the performance of MDC with time-varying channels. Kamnoonwatana et al. [115] present a redundancy allocation for a three-loop slice group MDC. The three-loop framework is proposed in [116] as a novel scheme that uses slice group coding tool proposed in H.264. In this framework, the rate distortion of side encoder can be controlled in a large dynamic range, so that redundancy can be easily managed with specific requirement of bitrate or side quality. The authors further build up two models to estimate the source and channel distortion. Source distortion model only estimates the side and central encoding distortion, whereas end-to-end distortion model considers channel statistics, error propagation, and



**FIGURE 8.6**

Packet loss performance comparison for the “News” sequence. (a) Packet loss: 5%, (b) packet loss: 10%, (c) packet loss: 15%, (d) packet loss: 20%. See [113].

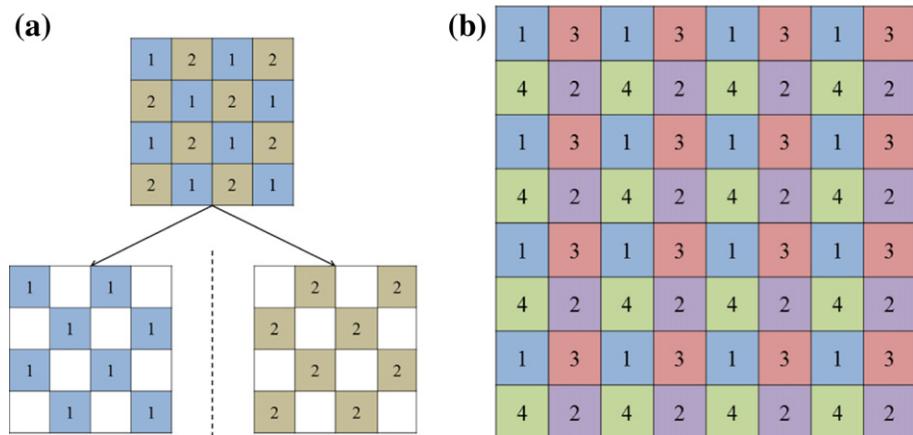
error concealment. The redundancy allocation algorithm assigns the best QP for encoder based on both source and channel distortion. Wei et al. [117] present an ARC scheme for a prediction-compensated polyphase MDC. Two aspects contribute to the ARC: quality and quantity control. The authors propose description-compensated redundancy generation scheme to produce effective redundancy information for quality control, and propose network-aware optimal redundancy allocation scheme for quantity control. Lin et al. [118] present an ARC scheme for MDC at MB level. An MB-level QP is assigned by the importance of the MB which is determined by the error propagation paths, the video contents, and the network status. This method is compatible with the baseline profile and extended profile of H.264/AVC. Tillo et al. [119] present an MDC method based on the redundant slice defined in the H.264/AVC standard. Two balanced descriptions are generated by an optimal redundancy allocation strategy that depends on the channel statistics and error propagation. This method is fully compatible with the H.264/AVC standard. Tillo et al. [120] present an MDC method for still images, based on optimal Lagrangian rate allocation. This scheme generates two balanced descriptions by splitting the coded blocks into two subsets with similar rate distortion and combining them, along with the redundancy allocation according to the network conditions. This method is compatible with JPEG 2000.

Recently, Ostergaard and Zamir revealed the connection between MDC and Delta-Sigma quantization (DSQ) [121]. The authors construct the MDC architecture based on DSQ oversampling and dithered lattice quantization. The authors also demonstrate that noise-shaping filter makes it possible to trade off central distortion for side distortion. This architecture is symmetric and does not need source splitting. Inspired by [121, 122] extends this principle to image coding that performs inter-block processing. In this method, DSQ is implemented with noise-shaping filters on image blocks first. Then coefficients and QPs are determined by rate-distortion optimization based on the statistics of the source image.

## 5.08.6 Network coding

Network coding (NC) as an information theory puzzle has become quite popular. For a given network topology, NC can increase the throughput and improve the network utilization by coding at intermediate nodes and transmitting the mixed packet to the next node [123]. An overview of network coding issues in wireless packet network can be found in [124]. The multicast problem using network coding is addressed in [125]. In practical network coding (PNC), the data stream is divided into generations, each of which is grouped into  $N$  packets. At the source, the packets from the same generation are linearly combined using random coefficients [126]. A linearly dependent description does not provide any additional information to the user. MDC produces correlated descriptions of the source data and avoids hierarchy between data layers. This property of MDC offers a typical solution for image/video transmission with channel diversity offered by wireless ad hoc networks [127].

Considering that MDVQ based on lattices has a better coding efficiency and easily extends to more than two descriptions than MDSQ, a joint MDC and NC scheme (MDC/NC) is described in [127] that uses three description lattice VQs for image coding. It uses NC combined with geographic routing. The experiments show that, MDC/NC achieve better decoded image quality and significantly lower failure rate with less energy consumption compared to the MDC with routine (MDC/R) scheme and the single description coding with NC (SDC/NC) scheme [127].

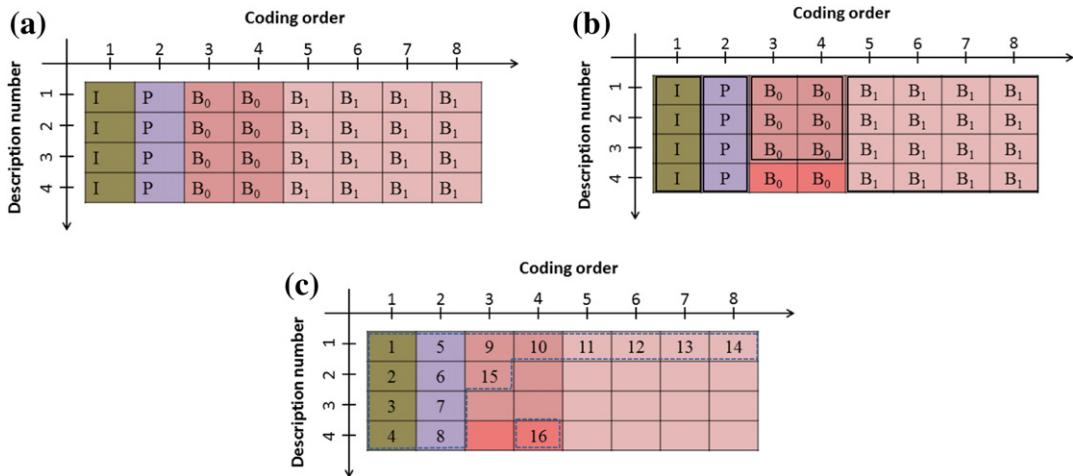
**FIGURE 8.7**

MDC subsampling schemes. (a) Quincunx subsampling and (b) polyphase subsampling. See [128].

MDC combined with NC has been used for image transmission over multiple links in a mesh network [128]. Two types of spatial subsampling schemes, namely Quincunx and Polyphase, are implemented along with a simple, yet effective use of NC to have a protected path formed by bitwise operations of data bitstreams. Two subsampling schemes are shown in Figure 8.7. The details of the experimental results are provided in [128].

In [126], it is shown that the number of linearly independent descriptions (LIDs) obtained by the receiver is almost directly proportional to its max-flow value. With some scenarios such as link and node failures, the receiver may not obtain as many LIDs as it should based on its max-flow capacity. The use of MDC reduces the effect of lower number of descriptions providing the user with a lower rate, but a satisfactorily decodable video content. In [126], a formulation of MD integrated with PNC (MD-PNC) is presented. This method is evaluated with and without QoS constraints. The QoS constraints impose maintaining certain average video quality in terms of PSNR, which can be mapped to a minimum rate constraint. Simulations done with the Network Simulator NS-2 for three types of users, namely “Low rate,” “Middle rate,” and “Premium,” show that MD-PNC with QoS constraints has a significant power over resources, ensuring the QoS to premium users even without the knowledge of the entire network; simultaneously optimizing the remaining resources for the other users [126].

To address the problem of the delay introduced by buffering before decoding in a real-time streaming scenario, a framework has been proposed in [129] combining expanding window network coding (EWNC), MDC, and a novel rate-distortion optimized (RDO) scheduling algorithm. The key concept behind EWNC is to increase the size of the “coding window”: the set of packets in the generation that may appear in combination vectors. This method provides instant decodability of packets using Gaussian elimination at the receiver side. Due to this property, EWNC is preferable over PNC [129]. The joint use of EWNC and MDC is shown to provide a robust video delivery over an unreliable wireless network,

**FIGURE 8.8**

Four-description scheduling framework. (a) A four-description and 8-frame window hierarchical B-frame GOP; (b) clustering of MD-GOP; (c) a possible schedule for first 16 packets. See [129].

without any need for centralized control for feedback channel. To facilitate this, the design of RDO method is proposed; which selects the video packet to be added at each sending opportunity in the way to maximize the expected video quality perceived by the receiver [129]. An example of four-description group-of-picture (GOP) structure using hierarchical B pictures is shown in Figure 8.8a. A clustering approach involving classification of the encoded video frames is proposed to improve the diversity. At each sending opportunity, a cluster is chosen among all clusters with compatible prediction levels by minimizing a cost function. Then, within the cluster, each sender chooses randomly one frame and schedules for transmission.

An example of a MD-GOP clustering is shown in Figure 8.8b, where frames marked by the same color belong to the same cluster. A possible schedule is presented in Figure 8.8c, where the numbers indicate the order in which the frame is included in the coding window and the dashed border identifies the frames selected for inclusion in the coding window at the 16th packet. The experiments show that the proposed approach consistently outperforms both EWNC used with a single description coding and EWNC used MDC without RDO. The detailed analysis is presented in [129].

A class of dynamic wireless networks that are self-organized in a mesh topology is that of mobile ad hoc networks (MANETs). Many random linear network coding (RLNC) based methods have been implemented for video transmission over MANET. The delay constraint produces a limitation of the generation size to the number of descriptions and this makes RLNC perform more poorly than the theoretical bound. A recent work described in [130] integrates MDC into NC to provide robustness and loosen the delay constraints. With the use of MDC the video quality is expected to be roughly proportional to the transmission data rate. A model is defined and a local optimization problem that maximizes the video quality perceived by the users, is considered in [130]. The experiments show that the

proposed technique performs on an average of about 2 dB better than RLNC. Also, the PSNR cumulative distribution function (cdf) shows that in case of RLNC, the distribution of PSNR is widespread showing inconsistency; whereas the proposed method can achieve consistency with almost all nodes performing in a close range. In general, for a typical scenario with stringent delay constraints, the optimization technique outperforms the random coefficient-based technique and provides a better quality video to the end users [130].

Dealing with client diversity is one of the main problems in multicasting video in a wireless LAN setting. Due to different channel characteristics, clients can receive different number of transmissions from the access point (AP). To overcome this issue, in [131], an approach combining multi-resolution coding (MRC) or MDC with inter-layer NC has been discussed. Firstly, a comparison between MRC and MDC without applying NC shows that, MDC outperforms MRC by 17% in terms of the number of decoded layers. Unlike MRC, receiving any number of descriptions contributes to the final quality of the video. With an idea of combining NC with MDC, two additional heuristic NC schemes that exploit the nature of MDC have been proposed.

The NC is done within each GOP. Let  $L$  be the number of layers/descriptions,  $N$  be the number of packets per layer, and  $X$  be the total number of transmissions the AP can have within the deadline of frames corresponding to  $N$ . Let  $R$  be the number of packets in a group to be considered for the same coding strategy. The basic *Canonical-L* scheme considers the triangular canonical form of strategies. *Canonical-(L + 1)* scheme considers  $(L + 1)$  ways of inter-layer coding, whereas *Canonical-(L + 4)* scheme considers  $(L + 4)$  ways of inter-layer coding. Complexity increases with each algorithm, e.g., for the typical values of  $(L, N, X, R) = (4, 8, 64, 4)$ , the above three schemes explore 969, 4845, and 245157 strategies, respectively. The design details of these schemes are presented in [131].

The experiments with  $L = 4$  show that NC combined with MDC improves the performance by 0–13.25%. It is noteworthy that, assuming MDC and MRC to have the same coding efficiency, NC benefits both schemes; but in the case of a single client the inherent advantage of MDC over MRC is lost when NC is applied to both schemes. For two clients, for  $L = 4$ , MDC with *Canonical-4* NC outperforms only MDC by 0–13.5%. However, at low loss rates, it underperforms only MDC. MDC with *Canonical-5* NC performs better than only MDC by a margin of 0–13% for almost all loss rates. MDC with *Canonical-5* NC outperforms MRC with *Canonical-4* NC in the range of only 3%, being consistent with the results for a single client. For more clients (4–6), MDC with NC performs better than the one without NC with a decline in the margin as the number of clients increases. The benefit of MDC with NC over MRC with NC remains negligible [131]. In all these experiments, the evaluation metric was the average number of decoded layers and not PSNR: frequently used in video coding literature. Taking into account the inter-description dependencies reducing coding efficiency in case of MDC, the authors conjecture the PSNR of the delivered video through MDC with NC to be lower than that of MRC with NC [131].

## 5.08.7 Stereoscopic 3D

Recently, there has been an interest in using MDC concepts for 3DTV and stereoscopic video. Two MDC techniques for stereoscopic video based on spatial and temporal redundancies are presented in [132, 133], both producing balanced descriptions and ensuring stereoscopic video reconstruction in the case of one-channel failure for the price of moderate coding redundancy. In the first, the *spatially scaled*

method, frames from the leading view are predicted by only frames of the same view while frames from the downsampled view are predicted by frames from the two views [133]. The other, the *multiple state* technique is based on temporal subsampling [84], in which motion-compensated prediction is performed separately in each description where left frames are predicted from preceding left frames, and right frames are predicted from preceding right frames or from the left frames corresponding to the same time instant [133]. An RRD analysis is done and the experiments show that the first approach (*spatially scaled*) performs better for sequences with lower inter-view correlation while the second approach (*multiple state*) performs better for sequences with higher inter-view correlation [132].

In [134], a scalable MDC scheme for 3D video has been developed. The proposed 3D scalable MDC in this paper begins with a standard SVC coder. The error resilience of the base layer of SVC is then enhanced using temporal MDC. The temporal MDC of the base layer is produced by using the MSVC approach as described in [84], which separates the even and odd frames of a sequence into two MDC streams [134]. In [135], a scalable MDC using motion vector coding for 3D stereoscopic video has been proposed. The proposed method improves the video quality while reducing redundancy and the system complexity [135].

### 5.08.8 Other applications

MDC has attracted considerable attention from some other new and evolving research areas. For watermarking applications, it is desirable to have some degree of correlation between the descriptions so that the reception of one or more of these will make it possible to detect the watermark with a certain confidence probability [136]. In the framework described in [136], a subset of descriptions is chosen for watermark insertion and is added together to form a watermarked signal. The method and experimental results for a DCT example are presented in [136]. In [137], a different approach based on iterative coding of multiple descriptions (ICMD) together with spread spectrum watermarking has been proposed to be used in noisy channels. An MDVQ-based image watermarking scheme has been proposed in [138, 139], where the MDVQ index assignments are modified suitable for watermark embedding and extraction. Another framework is presented in [140], which integrates an oblivious quantization index modulus modulation (QIMM) watermarking technique into MDC.

In an application of steganography, a covert communication scheme is proposed for a set partitioning in hierarchical trees (SPIHT) based image MDC [141]. First, the carrier image is subsampled into four subimages. The secret information is embedded in the subimages with a DWT and subsampling-based method. After the information is embedded the subimages are compressed by SPIHT, where one subimage corresponds to one SPIHT stream. Then the SPIHT streams are packed and transmitted through the packet-based channels individually. If there are packets lost, the lost packets can be substituted by the packets that reside in the other SPIHT streams [141].

In [142], the problem of delay-constrained multimedia applications over Cognitive Radio (CR) networks under Binomial primary traffic arrival is treated. An MDC based on the Priority Encoding Transmission (PET) method used on CR networks permits to improve the spectrum efficiency by sending only some descriptions instead of delivering the whole stream. Consequently, the spectrum resources involved in the transmission are minimized and the use of FEC mechanisms overcomes the interference losses [142].

---

### 5.08.9 Implementations and patents

Many MDC methods have been proposed in the patent literature. An MD joint source-channel (JSC) audio encoder has been implemented to encode  $n$  components of an audio signal over  $m$  transmission channels [143]. This audio encoder combines an MDTC with PAC. It is configured to select one or more transform parameters based on the signal characteristics. Another JSC encoder applied for image transmission has been implemented in [144]. This encoder forms vectors from the transform coefficients separated in space and frequency based on statistical redundancy allocation. These vectors can be selected to maximize the spatial separation between the transform coefficients. A method for using temporal prediction and motion-compensated prediction for MDC video has been implemented in [145]. The video encoder generates an output for each of the  $n$  channels using I-mode and P-mode encoding of MBs from every video frame. P-mode encoding is capable of generating at least  $n + 1$  prediction error signals. One of these can be utilized to reconstruct the original video regardless of the number of descriptions received by the decoder. This component is sent over each channel, whereas the rest  $n$  are sent over separate  $n$  channels. This method provides good results when the number of received descriptions is between 1 and  $n - 1$ . A method based on transforming two coefficients into two pairs of random variables is explained in [146]. One random variable from one pair is designed to have substantially equal energy as that of one random variable from another pair. This method included quantization of each pair of random variables followed by entropy coding generating encoded bitstreams. An application of an effective packetization of MDs is presented in [147]. In this method, for the given portion of signal, different descriptions are placed into packets such that at least a first description is placed into a first packet and a second into a second.

In [148], a method for handling off MD streaming media sessions between servers for fixed and mobile streaming media is presented. An adaptive method based on bitrates of set of coding parameters, channel characteristics, and end-to-end distortion model is implemented in [149]. The set of coding parameters that satisfies the available bitrate and minimizes the end-to-end distortion is chosen. An implementation of layered MDC is proposed in [150]. In this approach, the first layer included an initial part of a signal and its FEC code, the second part includes the next part with its FEC code. Cheng and Bauer [151] present an implementation of correlating and decorrelating transforms for MDC. This method uses fast Hadamard transforms to reduce the computational resources. In another implementation of MDC [152], a quantized bitstream is generated using a reference quantization step. Then, this bitstream is requantized using a first and a second quantization steps to generate two descriptions. A deployment of MDC for P2P video-on-demand streaming is presented in [153]. In this setup, if a sharing peer disconnects, the system searches for an alternate peer satisfying the bitrate and quality constraints. The system performance is claimed to improve as the number of descriptions increases, implying higher quality for the same network parameters. The design of a multi-level transform generating MDs is implemented in [154]. In this method, the descriptions are produced using a generation operation and variable support filters. Each operating point may have a corresponding error recovery process. Two types of processes are deployed with this setup. A four bitstream-based MDC method has been implemented with an error concealment process in [155]. Zhan [156] presents an MDC implementation that distributes the voice quality deteriorations to different parts and improves the user experience. A method for signaling and indicating track relationship in media files is presented in [157]. This, not directly an MDC implementation, however is useful for selecting a decodable track out of an MDC-track group.

### 5.08.10 Conclusion

In this chapter, we presented Multiple Description Coding as an effective solution to multimedia content delivery over unreliable network channel. MDC was proposed as a solution to the channel splitting problem for speech signals and developed theoretically as a generalized source coding problem. A promising framework for MDC was later established with the design of quantizers, transforms, and splitting mechanisms.

MDC has emerged as a powerful coding technique in numerous applications. It has been used as a robust image and speech coding method for more than a decade. It has been implemented in error resilient video coding as a valuable tool with applications in real-time video streaming, video conferencing and in scalable, P2P environments. MDC and network coding together have improved the QOS performance significantly. The most recent development of MDC clearly suggests the enormous potential that has yet to be exploited. With an ever-increasing use of wireless channels to communicate multimedia content; MDC may play a vital role in the design of future coding systems.

---

### List of Relevant Websites

EURASIP: <http://www.eurasip.org/index.php>

SPIE Digital Library: <http://spiedigitallibrary.org/>

ACM Digital Library: <http://dl.acm.org/>

IEEE Xplore: <http://ieeexplore.ieee.org/Xplore/home.jsp>

ELSEVIER: <http://www.elsevier.com/>

---

### Glossary

<b>Achievable quintuple</b>	a set of five parameters such that the rate coordinates lie in the rate region that is achievable for the upperbounds on distortion parameters
<b>Adaptive concealment</b>	a type of video error concealment method that makes use of certain video parameters to select the best concealment method adaptively out of the available choices of concealment methods
<b>Description MDSQ</b>	an encoded subsequence of a speech, image, or video source a scalar quantizer designed for encoding a source signal subject to a rate constraint, to form multiple descriptions to be sent across multiple channels
<b>Redundancy</b>	a measure of difference between the transmission rates of the actual source-coded bitstream and the bitstream that would achieve maximum compression of the source, with both bitstreams producing identical distortion upon reconstruction in case of error-free communication
<b>Spatial concealment</b>	a type of video error concealment method in which a lost part of a video is concealed using spatially neighboring reconstructed data, typically from the same frame
<b>Temporal concealment</b>	a type of video error concealment method in which a lost part of a video is concealed using temporally collocated reconstructed data, typically from one of the previously reconstructed frames

---

## References

- [1] Cisco visual networking index: forecast and methodology, 2011–2016, white paper, Cisco Systems Inc., Tech. Rep., 2012.
- [2] M. Yang, Multiple description video coding with adaptive error concealment (Ph.D. dissertation), Purdue University, West Lafayette, 2012.
- [3] A.M. Tekalp, Digital Video Processing, Prentice Hall, 1995.
- [4] T. Sikora, Trends and perspectives in image and video coding, Proc. IEEE 93 (1) (2005) 6–17.
- [5] V. Bhaskaran, K. Konstantinides, Image and Video Compression Standards: Algorithms and Architecture, Kluwer Academic Publishers, Boston, USA, 1997.
- [6] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005.
- [7] C.E. Shannon, Coding theorems for a discrete source with a fidelity criterion, Inst. Radio Eng. Int. Convention Rec. 7 (1959) 325–350.
- [8] A. Gamal, T. Cover, Achievable rates for multiple descriptions, IEEE Trans. Inf. Theory 28 (6) (1982) 851–857.
- [9] Y. Wang, A.R. Reibman, S. Lin, Multiple description coding for video delivery, Proc. IEEE 93 (1) (2005) 57–70.
- [10] V.K. Goyal, Multiple description coding: compression meets the network, IEEE Signal Process. Mag. 18 (5) (2001) 74–93.
- [11] J.K. Wolf, A.D. Wyner, J. Ziv, Source coding for multiple descriptions, Bell Syst. Tech. J. 59 (8) (1980) 1417–1426.
- [12] Z. Zhang, T. Berger, New results in binary multiple descriptions, IEEE Trans. Inf. Theory 33 (4) (1987) 502–521.
- [13] H.S. Witsenhausen, On source networks with minimal breakdown degradation, Bell Syst. Tech. J. 59 (6) (1980) 1083–1087.
- [14] H.S. Witsenhausen, A.D. Wyner, Source coding for multiple descriptions II: a binary source, Bell Syst. Tech. J. 60 (10) (1981) 2281–2292.
- [15] L. Ozarow, On a source-coding problem with two channels and three receivers, Bell Syst. Tech. J. 59 (10) (1980) 1909–1921.
- [16] N. Jayant, S. Christensen, Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure, IEEE Trans. Commun. 29 (2) (1981) 101–109.
- [17] N. Jayant, Subsampling of a DPCM speech channel to provide two self-contained half-rate channels, Bell Syst. Tech. J. 60 (4) (1981) 501–509.
- [18] A. Gamal, T. Cover, Information Theory of Multiple Descriptions Tech. Rep. 43, Department of Statistics, Stanford University, Stanford, CA, December 1980.
- [19] T. Berger, Z. Zhang, Minimum source degradation in binary source encoding, IEEE Trans. Inf. Theory 29 (6) (1983) 807–814.
- [20] R. Ahlswede, The rate-distortion region for multiple descriptions without excess rate, IEEE Trans. Inf. Theory 31 (6) (1985) 721–726.
- [21] R. Ahlswede, On multiple descriptions and team guessing, IEEE Trans. Inf. Theory 32 (4) (1986) 543–549.
- [22] V. Vaishampayan, Design of multiple description scalar quantizers, IEEE Trans. Inf. Theory 39 (3) (1993) 821–834.
- [23] Y. Wang, Q. Zhu, Error control and concealment for video communication: a review, Proc. IEEE 86 (5) (1998) 974–997.
- [24] V.A. Vaishampayan, J. Domaszewicz, Design of entropy-constrained multiple-description scalar quantizers, IEEE Trans. Inf. Theory 40 (1) (1994) 245–250.

- [25] V.A. Vaishampayan, J.C. Batlo, *Multiple description transform codes with an application to packetized speech*, in: Proceedings of the International Symposium on Information Theory, Trondheim, Norway, July 1994, p. 458.
- [26] V.A. Vaishampayan, Application of multiple description codes to image and video transmission over lossy networks, in: Proceedings of the 7th International Workshop Packet Video, Brisbane, Australia, 1996, pp. 55–60.
- [27] J.C. Batlo, V.A. Vaishampayan, *Asymptotic performance of multiple description transform codes*, *IEEE Trans. Inf. Theory* 43 (2) (1997) 703–707.
- [28] Y. Wang, M.T. Orchard, A.R. Reibman, Multiple description image coding for noisy channels by pairing transform coefficients, in: Proceedings of the IEEE First Workshop on Multimedia Signal Processing, Princeton, USA, June 1997, pp. 419–424.
- [29] M.T. Orchard, Y. Wang, V. Vaishampayan, A.R. Reibman, *Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms*, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Washington, USA, vol. 1, October 1997, pp. 608–611.
- [30] Y. Wang, M. Orchard, V. Vaishampayan, A. Reibman, Multiple description coding using pairwise correlating transforms, *IEEE Trans. Image Process.* 10 (3) (2001) 351–366.
- [31] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, R. Puri, Multiple description coding for video using motion compensated prediction, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Kobe, Japan, vol. 3, October 1999, pp. 837–841.
- [32] S.S. Hemami, Reconstruction-optimized lapped orthogonal transforms for robust image transmission, *IEEE Trans. Circ. Syst. Video Technol.* 6 (2) (1996) 168–181.
- [33] H.S. Malvar, D.H. Staelin, The LOT: transform coding without blocking effects, *IEEE Trans. Acoust. Speech Signal Process.* 37 (4) (1989) 553–559.
- [34] T.M. Cover, J.A. Thomas, *Elements of Information Theory*, Wiley, New York, USA, 1991.
- [35] C.E. Shannon, A mathematical theory of communications, *Bell Syst. Tech. J.* 27 (379–423) (1948) 623–656.
- [36] R. Zamir, Gaussian codes and shannon bounds for multiple descriptions, *IEEE Trans. Inf. Theory* 45 (7) (1999) 2629–2636.
- [37] R. Venkataramani, G. Kramer, V. Goyal, Multiple description coding with many channels, *IEEE Trans. Inf. Theory* 49 (9) (2003) 2106–2114.
- [38] Y. Wang, M. Orchard, A. Reibman, Optimal pairwise correlating transforms for multiple description coding, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Chicago, USA, vol. 1, October 1998, pp. 679–683.
- [39] A. Reibman, H. Jafarkhani, M. Orchard, Y. Wang, Performance of multiple description coders on a real channel, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Phoenix, Arizona, vol. 5, March 1999, pp. 2415–2418.
- [40] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, R. Puri, Multiple description video coding using motion compensated temporal prediction, *IEEE Trans. Circ. Syst. Video Technol.* 12 (3) (2002) 193–204.
- [41] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, Multiple description video using rate-distortion splitting, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Thessaloniki, Greece, vol. 1, October 2001, pp. 978–981.
- [42] V.K. Goyal, J. Kovacevic, Generalized multiple description coding with correlating transforms, *IEEE Trans. Inf. Theory* 47 (6) (2001) 2199–2224.
- [43] C. Tian, S. Hemami, A new class of multiple description scalar quantizer and its application to image coding, *IEEE Signal Process. Lett.* 12 (4) (2006) 329–332.
- [44] V.A. Vaishampayan, J.C. Batlo, Asymptotic analysis of multiple description quantizers, *IEEE Trans. Inf. Theory* 44 (1) (1998) 278–284.

- [45] V.A. Vaishampayan, N.J.A. Sloane, S.D. Servetto, Multiple description vector quantization with lattice codebooks: design and analysis, *IEEE Trans. Inf. Theory* 47 (5) (2001) 1718–1734.
- [46] J.H. Conway, N.J.A. Sloane, Voronoi regions of lattices, second moments of polytopes and quantization, *IEEE Trans. Inf. Theory* 28 (1982) 211–226.
- [47] J.H. Conway, N.J.A. Sloane, Fast quantization and decoding algorithms for lattice quantizers and codes, *IEEE Trans. Inf. Theory* 28 (1982) 227–232.
- [48] S.N. Diggavi, N.J. Sloane, V.A. Vaishampayan, Asymmetric multiple description lattice vector quantizers, *IEEE Trans. Inf. Theory* 48 (1) (2002) 174–191.
- [49] H. Jafarkhani, V. Tarokh, Multiple description trellis-coded quantization, *IEEE Trans. Commun.* 47 (6) (1999) 799–803.
- [50] H. Jafarkhani, V. Tarokh, Multiple description trellis-coded quantization, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Chicago, USA, vol. 1, October 1998, pp. 669–673.
- [51] J.G.D. Forney, The viterbi algorithm, *Proc. IEEE* 61 (3) (1973) 268–278.
- [52] T. Lam, G. Abusleman, L. Karam, Multiple description channel-optimized trellis-coded quantization, *IEEE Transactions on Acoustics Speech Signal Processing (ICASSP)*, Istanbul, Turkey, vol. 5, June 2000, pp. 2645–2648.
- [53] J.G. Proakis, D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, Prentice-Hall, New Jersey, USA, 1996.
- [54] R. Arean, J. Kovacevic, V.K. Goyal, Multiple description perceptual audio coding correlating transforms, *IEEE Trans. Speech Audio Process.* 8 (2) (2000) 140–145.
- [55] W. Jiang, A. Ortega, Multiple description speech coding for robust communication over lossy packet networks, in: Proceedings of the International Conference on Multimedia and Expo (ICME), New York City, USA, vol. 1, July–August 2000, pp. 444–447.
- [56] Z. Xin, J. Arrowood, A. Moreno, M. Clements, Multiple description coding for recognizing voice over IP, in: Proceedings of 10th Digital Signal Processing Workshop and the 2nd Signal Processing Education Workshop, Pine Mountain, USA, October 2002, pp. 383–386.
- [57] L. Dong, B. Wah, LSP-based multiple description coding for real-time low bit-rate voice transmissions, in: Proceedings of the International Conference on Multimedia and Expo (ICME), Lusanne, Switzerland, vol. 2, August 2002, pp. 597–600.
- [58] L. Dong, B. Wah, LSP-based multiple-description coding for real-time low bit-rate voice over IP, *IEEE Trans. Multimedia* 7 (1) (2005) 167–178.
- [59] H. Dong, A. Gersho, J. Gibson, V. Cuperman, A multiple description speech coder based on AMR-WB for mobile ad-hoc networks, *IEEE Transactions on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal, Canada, vol. 1, May 2004, pp. 277–280.
- [60] G. Schuller, J. Kovacevic, V.K. Goyal, Robust low-delay audio coding using multiple descriptions, *IEEE Trans. Speech Audio Process.* 13 (5) (2005) 1014–1024.
- [61] S.D. Voran, A multiple description speech coding using structural dual vector quantizers, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Philadelphia, USA, vol. 1, March 2005, pp. 129–132.
- [62] M.D. Kwong, S. Cherkaoui, R. Lefebvre, Multiple description and multi-path routing for robust voice transmission over ad hoc networks, in: Proceedings of the IEEE Wireless and Mobile Computing, Networking and Communications, 2006 (WiMob'2006), Vancouver, Canada, June 2006, pp. 262–267.
- [63] P. Yahampath, P. Rondeau, Multiple-description predictive-vector quantization with applications to low bit-rate speech coding over networks, *IEEE Trans. Audio Speech Lang. Process.* 15 (3) (2007) 749–755.
- [64] S. Hojjat, K. Sadri, S. Shirani, Multiple description coding of audio using phase scrambling, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Hannover, June 2008, pp. 153–156.

- [65] M. Larsen, M. Christensen, S. Jensen, A multiple description quantization of sinusoidal parameters, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Las Vegas, USA, March–April 2008, pp. 189–192.
- [66] J. Jensen, M.G. Christensen, M.H. Jensen, S.H. Jensen, T. Larsen, Multiple description spherical quantization of sinusoidal parameters with repetition coding of the amplitudes, in: The 43rd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, USA, November 2009, pp. 385–389.
- [67] J. Ostergaard, D.E. Quevedo, J. Jensen, Low delay moving-horizon multiple-description audio coding for wireless hearing aids, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Taipei, Taiwan, April 2009, pp. 21–24.
- [68] J. Ostergaard, D.E. Quevedo, J. Jensen, Real-time perceptual moving-horizon multiple-description audio coding, *IEEE Trans. Signal Process.* 59 (9) (2011) 4286–4299.
- [69] W. Hoseok, A. Ito, Y. Suzuki, Multiple description coding for wideband audio signal transmission, in: Proceedings of the IEEE International Conference on Network Infrastructure and Digital Content 2009, Beijing, China, September 2009, pp. 769–773.
- [70] X. Zheng, C. Ritz, Packet loss protection for interactive audio object rendering: a multiple description approach, in: Fourth International Workshop on Quality of Multimedia Experience (QoMEX) 2012, Yarra Valley, Australia, July 2012, pp. 68–73.
- [71] V. Goyal, J. Kovacevic, R. Arean, M. Vetterli, Multiple description transform coding of images, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Chicago, USA, vol. 1, October 1998, pp. 674–678.
- [72] D. Chung, Y. Wang, Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms, *IEEE Trans. Circ. Syst. Video Technol.* 9 (6) (1999) 895–910.
- [73] S. Servetto, K. Ramchandran, V. Vaishampayan, K. Nahrstedt, Multiple description wavelet based image coding, *IEEE Trans. Image Process.* 9 (5) (2000) 813–826.
- [74] M. Pereira, M. Antonini, M. Barlaud, Channel adapted multiple description coding scheme using wavelet transform, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Rochester, USA, vol. 2, September 2002, pp. II-197–II-200.
- [75] S. Shirani, M. Gallant, F. Kosseintini, Multiple description image coding using pre-and post-processing, in: Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, USA, April 2001, pp. 35–39.
- [76] N. Franchi, M. Fumagalli, R. Lancini, Flexible redundancy insertion in a polyphase down sampling multiple description image coding, in: Proceedings of the IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, vol. 2, August 2002, pp. 605–608.
- [77] L. Wang, M. Swamy, M. Ahmad, Multiple description image coding using pixel interleaving and wavelet transform, The 45th Midwest Symposium on Circuits and Systems 2002, MWSCAS-2002, vol. 2, August 2002, pp. II-235–II-238.
- [78] A. Ashwin, K. Ramakrishnan, S. Srinivasan, A multiple description method for wavelet based image coding, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Rochester, USA, vol. 2, September 2002, pp. 709–712.
- [79] I.V. Bajic, J.W. Woods, Domain-based multiple description coding of images and video, *IEEE Trans. Image Process.* 12 (10) (2003) 1211–1225.
- [80] J. Chen, C. Cai, S. Mitra, Unequal loss protected multiple description subband coding, in: Proceedings of the IEEE International Conference on Communications, Circuits and Systems, 2004 (ICCCAS-2004), Chengdu, China, vol. 2, June 2004, pp. 919–922.
- [81] C. Cai, J. Chen, R. Ding, Significant coefficient decomposition based stack X-tree multiple description coding, in: Proceedings of the IEEE International Conference on Neural Networks and Signal Processing, 2003 (ICNNSP2003), Nanjing, China, vol. 2, December 2003, pp. 1181–1184.

- [82] K. Sadri, S. Shirani, Multiple description coding of images using phase scrambling, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Montreal, Canada, vol. III, May 2004, pp. 41–44.
- [83] C. Tian, S. Shirani, Visually optimized multiple description image coding, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France, vol. II, May 2006, pp. 1–4.
- [84] J.G. Apostolopoulos, Error-resilient video compression through the use of multiple states, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Vancouver, Canada, vol. 3, September 2000, pp. 352–355.
- [85] N. Franchi, M. Fumagalli, R. Lancini, S. Tubaro, Multiple description video coding for scalable and robust transmission over IP, *IEEE Trans. Circ. Syst. Video Technol.* 15 (3) (2005) 321–334.
- [86] N.V. Boulgouris, K.E. Zachariadis, A.N. Leontaris, M.G. Strintzis, Drift-free multiple description coding of video, in: Proceedings of the IEEE Fourth Workshop on Multimedia Signal Processing, Cannes, France, October 2001, pp. 105–110.
- [87] C. Kim, S. Lee, Multiple description coding of motion fields for robust video transmission, *IEEE Trans. Circ. Syst. Video Technol.* 11 (9) (2001) 999–1010.
- [88] Y. Wang, S. Lin, Error-resilient video coding using multiple description motion compensation, *IEEE Trans. Circ. Syst. Video Technol.* 12 (6) (2002) 438–452.
- [89] V.V. Vaishampayan, S. John, Balanced interframe multiple description video compression, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Kobe, Japan, vol. 3, October 1999, pp. 812–816.
- [90] S. Regunathan, K. Rose, Efficient prediction in multiple description video coding, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Vancouver, Canada, vol. 1, September 2000, pp. 1020–1023.
- [91] X. Tang, A. Zakhor, Matching pursuits multiple description coding for wireless video, *IEEE Trans. Circ. Syst. Video Technol.* 12 (6) (2002) 566–575.
- [92] K. Matty, L. Kondi, Balanced multiple description video coding using optimal partitioning of the DCT coefficients, *IEEE Trans. Circ. Syst. Video Technol.* 15 (7) (2005) 928–934.
- [93] T. Wiegand, G. Sullivan, G. Bjontegaard, A. Luthra, Overview of the H.264/AVC video coding standard, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 560–576.
- [94] T. Tillo, G. Olmo, Data-dependent pre-and postprocessing multiple description coding of images, *IEEE Trans. Image Process.* 16 (5) (2007) 1269–1280.
- [95] M. Gallant, S. Shirani, F. Kossentini, Standard-compliant multiple description video coding, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Thessaloniki, Greece, vol. 1, October 2001, pp. 946–949.
- [96] Z. Wei, C. Cai, K.-K. Ma, A novel H.264-based multiple description video coding via polyphase transform and partial prediction, in: Proceedings of the International Symposium on Intelligent Signal Processing and Communications, Yonago, Japan, vol. 1, 2006, pp. 151–154.
- [97] S. Shirani, Content-based multiple description image coding, *IEEE Trans. Multimedia* 8 (2) (2006) 411–419.
- [98] D. Wang, N. Canagarajah, D. Redmill, D. Bull, Multiple description video coding based on zero padding, in: Proceedings of the International Symposium on Circuits and Systems 2004, Vancouver, Canada, vol. 2, May 2004, pp. 205–208.
- [99] G. Zhang, R.L. Stevenson, Efficient error recovery for multiple description video coding, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Singapore, vol. 2, October 2004, pp. 829–832.
- [100] M.V. der Schaar, D.S. Turaga, Multiple description scalable coding using wavelet-based motion compensated temporal filtering, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Barcelona, Spain, vol. 3, September 2003, pp. 489–492.

- [101] E. Akyol, A.M. Tekalp, M.R. Civanlar, Scalable multiple description video coding with flexible number of descriptions, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Genoa, Italy, vol.3, September 2005, pp. 712–715.
- [102] E. Akyol, A. Tekalp, M. Civanlar, A flexible multiple description coding framework for adaptive peer-to-peer video streaming, *IEEE J. Sel. Top. Signal Process.* 1 (2) (2007) 231–245.
- [103] C. Zhu, M. Liu, Multiple description video coding based on hierarchical B pictures, *IEEE Trans. Circ. Syst. Video Technol.* 19 (4) (2009) 511–521.
- [104] R. Puri, K. Ramchandran, Multiple description source coding using forward error correction codes, Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, USA, vol. 1, October 1999, pp. 342–346.
- [105] R. Bernardini, M. Durigon, R. Rinaldo, L. Celetto, A. Vitali, Polyphase spatial subsampling multiple description coding of video streams with h264, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Singapore, vol. 5, October 2004, pp. 3213–3216.
- [106] J.R. Ohm, Three-dimensional subband coding with motion compensation, *IEEE Trans. Image Process.* 3 (5) (1994) 559–571.
- [107] C. Hsiao, W. Tsai, Hybrid multiple description coding based on H.264, *IEEE Trans. Circ. Syst. Video Technol.* 20 (1) (2010) 76–87.
- [108] M. Yang, M.L. Comer, E.J. Delp, A four-description MDC for high loss-rate channels, in: Proceedings of the Picture Coding Symposium, Nagoya, Japan, December 2010.
- [109] N. Khanna, F. Zhu, M. Bosch, M. Yang, M. Comer, E.J. Delp, Information theory inspired video coding methods: truth is sometimes better than fiction, in: Proceedings of the Third Workshop on Information Theoretic Methods in Science and Engineering, Tampere, Finland, August 2010.
- [110] M. Yang, Y. He, F. Zhu, M. Bosch, M. Comer, E.J. Delp, Video coding: death is not near, in: Proceedings of the 53rd International Symposium ELMAR, Zadar, Croatia, 2011, pp. 85–88.
- [111] M. Yang, M.L. Comer, E.J. Delp, An adaptable spatial-temporal error concealment method for multiple description coding based on error tracking, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, September 2011.
- [112] M. Yang, M.L. Comer, E.J. Delp, Macroblock-level adaptive error concealment methods for MDC, in: Proceedings of the Picture Coding Symposium 2012, Krakow, Poland, May 2012.
- [113] N. Gadgil, M. Yang, M.L. Comer, E.J. Delp, Adaptive error concealment for multiple description video coding using motion vector analysis, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Orlando, USA, October 2012.
- [114] R. Yang, Efficient inter-layer motion compensation and error resilience for spatially scalable video coding (Ph.D. dissertation), Purdue University, West Lafayette, 2009.
- [115] N. Kamnoonwatana, D. Agrafiotis, C.N. Canagarajah, Flexible adaptive multiple description coding for video transmission, *IEEE Trans. Circ. Syst. Video Technol.* 22 (1) (2012) 1–11.
- [116] D. Wang, N. Canagarajah, D. Bull, Slice group based multiple description video coding with three motion compensation loops, in: IEEE International Symposium on Circuits and Systems 2005 (ISCAS 2005), vol. 2, 2005, pp. 960–963.
- [117] Z. Wei, K. Ma, C. Cai, Prediction-compensated polyphase multiple description image coding with adaptive redundancy control, *IEEE Trans. Circ. Syst. Video Technol.* 22 (3) (2012) 465–478.
- [118] C. Lin, T. Tillo, Y. Zhao, B. Jeon, Multiple description coding for H.264/AVC with redundancy allocation at macro block level, *IEEE Trans. Circ. Syst. Video Technol.* 21 (5) (2011) 589–600.
- [119] T. Tillo, M. Grangetto, G. Olmo, Redundant slice optimal allocation for H.264 multiple description coding, *IEEE Trans. Circ. Syst. Video Technol.* 18 (1) (2008).

- [120] T. Tillo, M. Grangetto, G. Olmo, Multiple description image coding based on lagrangian rate allocation, *IEEE Trans. Image Process.* 16 (3) (2007) 673–683.
- [121] J. Ostergaard, R. Zamir, Multiple-description coding by dithered delta-sigma quantization, in: Proceedings of Data Compression Conference, March 2007, pp. 63–72.
- [122] Y. Fan, J. Wang, J. Sun, Multiple description image coding based on delta-sigma quantization with rate-distortion optimization, *IEEE Trans. Image Process.* 21 (9) (2012) 4304–4309.
- [123] R. Ahlswede, N. Cai, S. Li, R. Yeung, Network information flow, *IEEE Trans. Inf. Theory* 46 (2000) 1204–1215.
- [124] D. Lun, M.M. dard, R. Koetter, Efficient opereation of wireless packet networks using network coding, in: Proceedings of the International Workshop on Convergent Technologies (IWCT), Oulu, Finland, June 2005, pp. 1–5.
- [125] D. Lun, N. Ratnakar, M.M. dard, R. Koetter, D. Karger, T. Ho, E. Ahmed, F. Zhao, Minimum-cost multicast over coded packet networks, *IEEE Trans. Inf. Theory* 52 (6) (2006) 2608–2623.
- [126] A. Ramasubramonian, J. Woods, Multiple description coding and practical network coding for video multic平, *IEEE Signal Process. Lett.* 17 (3) (2010) 265–268.
- [127] Y. Xu, C. Zhu, Joint multiple description coding and network coding for wireless image multicast, in: Proceedings of the IEEE International Conference on Image and Graphics, Xi'an, China, September 2009, pp. 819–823.
- [128] H. Maza'ar, H. Elmahdy, Multiple description image network coding against single link failure in mesh networks, in: Proceedings of the IEEE International conference on Internet Multimedia Services Architecture and Applications, IMSAA-2009, Bangalore, India, December 2009, pp. 1–5.
- [129] C. Greco, I. Nemoianu, M. Cagnazzo, B. Pesquet-Popescu, A network coding scheduling for multiple description video streaming over wireless networks, in: Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, August 2012, pp. 1915–1919.
- [130] I. Nemoianu, C. Greco, M. Cagnazzo, B. Pesquet-Popescu, A framework for joint multiple description coding and network coding over wireless ad-hoc networks, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, March 2012, pp. 2309–2312.
- [131] R. Gandhi, M. Yang, D. Koutsonikolas, Y. Hu, M. Comer, A. Mohamed, C. Wang, The impact of inter-layer network coding on the relative performance of MRC/MDC WiFi media delivery, in: Proceedings of the 21st International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2011), Vancouver, Canada, June 2011, pp. 27–32.
- [132] A. Norkin, A. Aksay, C. Bilen, G. Akar, A. Gotchev, J. Astola, Schemes for multiple description coding of stereoscopic video, in: Proceedings of the 2006 International Workshop on Multimedia Content Representation, Classification and Security, Istanbul, Turkey, September 2006, pp. 730–737.
- [133] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. Akar, G. Triantafyllidis, A. Koz, Coding algorithm for 3DTV—a survey, *IEEE Trans. Circ. Syst. Video Technol.* 17 (11) (2007) 1606–1621.
- [134] H. Karim, C. Hewage, S. Worrall, A. Kondoz, Scalable multiple description video coding for stereoscopic 3D, *IEEE Trans. Consum. Electron.* 54 (2) (2008) 745–752.
- [135] S. Adedoyin, W. Fernando, A. Kondoz, Scalable MDC for 3D stereoscopic video using motion vector encoding, in: Proceedings of the International Conference on Multimedia and Expo (ICME), Singapore, July 2010, pp. 1718–1723.
- [136] R. Chandramouli, B. Graubard, C. Richmond, A multiple description framework for oblivious watermarking, in: Proceedings of the SPIE Security and Watermarking of Multimedia Contents III, vol. 4314, 2001.
- [137] Y. Hsia, C. Chang, J. Liao, Multiple-description coding for robust image watermarking, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Singapore, October 2004, pp. 2163–2166.

- [138] J. Pan, Y. Hsin, H. Huang, K. Huang, Robust image watermarking based on multiple description vector quantisation, *Electron. Lett.* 40 (22) (2004).
- [139] S. Chu, H. Hsin, H. Huang, K. Huang, J. Pan, Multiple description watermarking for lossy network, in: *Proceedings of the IEEE International Symposium on Circuits and Systems 2005 (ISCAS 2005)*, Kobe, Japan, vol. 4, May 2005, pp. 3990–3993.
- [140] M. Day, S. Lee, I. Jou, Multiple description watermarking based on quantization index modulus modulation, *J. Inf. Sci. Eng.* 23 (6) (2007) 1785–1800.
- [141] C. Lin, J. Pan, K. Huang, A covert communication scheme for a spih based image multiple description coding system, in: *The Third International Conference on Innovative Computing Information and Control (ICICIC-2008)*, Dalian, China, June 2008, p. 440.
- [142] A. Chaoub, E. Ibn-Elhaj, Multiple description coding for cognitive radio networks under secondary collision errors, in: *Proceedings of 16th IEEE Mediterranean Electrotechnical Conference (MELECON)*, 2012, Tunisia, March 2012, pp. 27–30.
- [143] R. Aeron, V. Goyal, J. Kovacevic, Multiple description transform coding of audio using optimal transforms of arbitrary dimension, Patent US 6,253,185 B1, June 26, 2001.
- [144] V. Goyal, J. Kovacevic, M. Vetterli, Multiple description transform coding of images using optimal transforms of arbitrary dimension, Patent US 6,330,370 B2, December 11, 2001.
- [145] M. Orchard, H. Jafarkhani, A. Reibman, Y. Wang, Method and apparatus for accomplishing multiple description coding for video, Patent US 6,556,624 B1, April 29, 2003.
- [146] H. Jafarkhani, M. Orchard, A. Reibman, Y. Wang, Multiple description coding communication system, Patent US 6,823,018 B1, November 23, 2004.
- [147] V. Goyal, J. Kovacevic, F. Masson, Method and apparatus for wireless transmission using multiple description coding, Patent US 6,983,243 B1, January 3, 2006.
- [148] J.G. Apostolopoulos, S. Basu, G. Cheung, R. Kumar, S. Roy, W.-t Tan, S.J. Wee, T. Wong, B. Shen, Method for handling off multiple description streaming media sessions between servers in fixed and mobile streaming media systems, Patent US 6,996,618 B2, February 7, 2006.
- [149] S. Lin, A. Vetro, Y. Wang, Adaptive error-resilient video encoding using multiple description motion compensation, Patent US 7,106,907 B2, September 12, 2006.
- [150] P.A. Chou, V.N. Padmanabhan, H. Wang, Layered multiple description coding, Patent US 7,426,677 B2, September 16, 2008.
- [151] C. Cheng, C. Bauer, Correlating and decorrelating transforms for multiple description coding system, Patent US 7,536,299 B2, May 19, 2009.
- [152] A.C. Irvine, V.R. Raveendran, Apparatus and method for multiple description encoding, Patent US 7,561,073 B2, July 14, 2009.
- [153] S.S. Panwar, K.W. Ross, Y. Wang, On demand peer-to-peer video streaming with multiple description coding, Patent US 7,633,887 B2, December 15, 2009.
- [154] M. Paniconi, J. James J. Carrig, Z. Miao, Variable support robust transform for multiple description coding, Patent US 7,817,869 B2, October 19, 2010.
- [155] M. Cancemi, A.L. Vitali, Method and system for multiple description coding and computer program product therefor, Patent US 7,991,055 B2, August 2, 2011.
- [156] W. Zhan, Method, apparatus and system for multiple-description coding and decoding, Patent US 8,279,947 B2, October 2, 2012.
- [157] Y.-K. Wang, M. Hannuksela, I. Bouazizi, Apparatus and method for indicating track relationships in media files, Patent US 8,365,060 B2, January 29, 2013.

# Video Error Concealment

9

Dimitris Agrafiotis

*Visual Information Laboratory, Department of Electrical and Electronic Engineering, University of Bristol, UK  
Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK*

---

## Nomenclature

ARQ	automatic repeat request
AVC	advanced video codec
BI	bidirectional interpolation
BMA	block matching algorithm
BME	boundary matching error
BNM	best neighborhood matching
DI	directional interpolation
EBME	external boundary matching error
EC	error concealment
EECMS	enhanced error concealment with mode selection
FMO	flexible macroblock ordering
HEVC	high efficiency video codec
IDR	instantaneous decoder refresh
MB	macroblock
MCTA	motion compensated temporal activity
MPEG	motion pictures experts group
MS	mode selection
MV	motion vector
SA	spatial activity
SAD	sum of absolute differences
SEC	spatial error concealment
TEC	temporal error concealment

---

### 5.09.1 Introduction

Error concealment provides the last line of defence for a video delivery system in its battle against transmission errors. In the context of video coding standards, it is a non-normative post-processing

operation that takes place after the decoding of the current frame and prior to its display, or in the case of whole frame loss, after a missing frame has been detected and prior to decoding the next frame.

Encoded video bitstreams, because of the compression methods employed, are generally highly sensitive to channel errors. Due to the widespread use of prediction methods (e.g., for motion compensation, for differential DC transform coefficient coding, and for motion vector coding) and variable length coding (e.g., in entropy coding for motion vectors and residual data), a single uncorrected channel error can propagate to corrupt adjacent spatiotemporal regions of a video signal. The video transmission chain therefore usually includes forward error correction coding in order to protect video packets prior to sending them over a channel to the receiver.

When the number of errors exceeds the correction capabilities of the error correction code employed, many communication systems will invoke an auto-repeat request (ARQ) process, whereby the missing packets are retransmitted. ARQ methods are not however applicable in multicast transmission, where clients experience different channel conditions. Even in unicast transmission, the number of ARQs is often limited because of real-time rendering or buffering constraints. When these limits are exceeded, packets are not received correctly and are usually not propagated to the application layer leading to missing blocks of pixels in the decoded frame or whole frame losses. The extent of loss will depend on the packetization applied at the transmitter and the severity of the channel errors [1]. Missing packets can also occur when delays exceed the limits imposed by the delay-sensitive nature of video applications, whereby video packets have to be displayed at certain time intervals or dropped otherwise.

To avoid a large drop in received video quality due to intra- and inter-frame propagation of such errors, incorporation of error resilience features at the encoder and the use of error concealment at the decoder are necessary [2–5]. Error resilience is the process of adding redundancy to the video stream during encoding for the purpose of preventing severe video quality degradation at the decoder in the presence of errors. A typical example is the use of multiple slices within a frame. Slices are independently decodable units that consist of a number of coding units or macroblocks<sup>1</sup> (e.g., a row). They add redundancy to the coded stream in the form of headers but also by setting allowable limits for intra-frame prediction of pixel values and side information. Their use prevents intra-frame error propagation. Without them a missing packet would result in a whole frame loss and create very difficult (if not impossible) conditions for the error concealment module at the decoder. Other such error resilience tools include redundant slices (data repeated at a lower fidelity), data partitioning followed by unequal error protection (separation of motion vectors, intra data, and residual), flexible macroblock ordering (macroblock interleaving patterns), and instantaneous decoder refresh frames (IDR frames) [6]. Error resilience features are not covered in detail in this chapter, but the reader is referred to [7].

In contrast to error resilience, which helps to protect the bitstream against loss, error concealment (EC) is a receiver-based process which estimates the content lost during transmission and replaces it prior to rendering. To accomplish this recovery, concealment methods must exploit the correlations that exist between a damaged macroblock (MB) and its adjacent neighbors in the same or previous frame(s). Two primary types of concealment exist: temporal error concealment (TEC) and spatial error concealment (SEC). Temporal error concealment methods estimate missing motion vectors (MVs) which they then use for motion compensated temporal replacement of the lost MBs, while spatial error concealment

---

<sup>1</sup>Without loss of generality, the term macroblock is used throughout this chapter.

methods base their predictions of missing data on spatially adjacent macroblocks, and normally employ some form of interpolation.

This chapter focuses on the characterization and design of video error concealment methods that deal with missing macroblocks. Given that error concealment is a post-processing operation, the methods are presented in a standard-independent way—however they are applicable to all currently used block-based codecs including MPEG-2, H.264/AVC, or the latest HEVC video coding standard [8]. Methods for H.264 extensions such as SVC [9] and multi-view [10] have also been reported in the literature but are not discussed herein. TEC methods that deal with the concealment of whole missing frames have also been reported in [11,12] among others.

Despite the very large number of methods reported in the literature, most of them follow similar steps with relatively few variations. In order to guide the reader through these methods and to provide a path to building an error concealment module/method, we have followed a recipe book format for the three primary concealment stages—Spatial Error Concealment, Temporal Error Concealment, and Mode Selection. All the necessary ingredients are listed first, followed by a description of how these are employed, including any variations to the recipe. The important tools necessary for each method are then described, followed by a performance analysis. The reader should thus have sufficient information to construct an error concealment method with good performance. Throughout, performance comparisons are included to demonstrate the relative strengths and advantages of some of the methods described. Complexity issues are also briefly discussed in the final section.

## 5.09.2 Spatial error concealment (SEC)

### 5.09.2.1 Ingredients

- Selected neighboring pixel values or blocks of pixels.
- One or more interpolation/replacement methods.
- A switching method (optional—if more than one replacement method is used).

### 5.09.2.2 Method

#### 5.09.2.2.1 Boundary pixel selection

Having identified the missing macroblocks, we must first decide which boundary pixels from adjacent MBs will take part in the concealment process. Boundary pixels belonging to previously concealed MBs are normally considered unreliable and excluded from the following steps. However, in certain situations where there is a lack of sufficient reliable neighbors, their use may be necessary.

#### 5.09.2.2.2 Interpolation of lost pixels

An interpolation method, using the selected boundary pixels, must be applied in order to replace the missing pixels. The method of Salama et al. [13] uses *bilinear interpolation* (BI) to replace the missing pixels with weighted averages of the boundary pixels. This approach will result in smooth approximations of the missing pixels/MBs. In smooth or even in highly textured areas, such an approach will give adequate results. However, in the presence of edges, it can create disturbing artifacts (Figure 9.1).

**FIGURE 9.1**

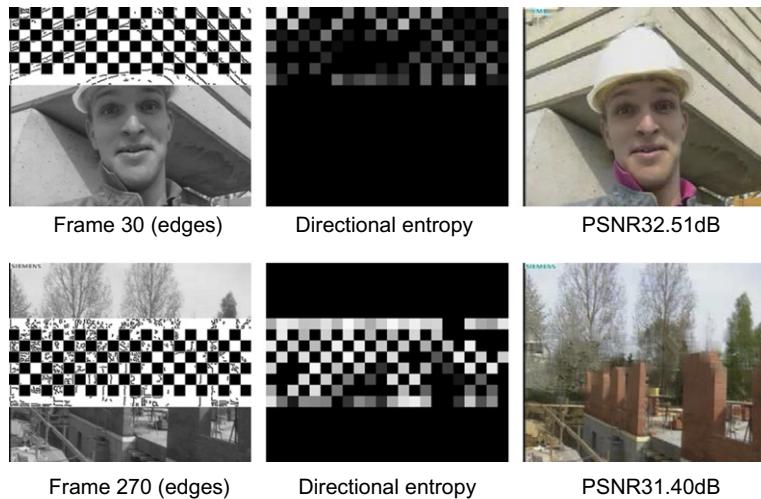
Concealment of the missing MBs shown in (a) (“foreman” frame 30) using DI (b) and BI (c).

To avoid the problems discussed above, the direction of edges passing through the missing macroblock(s) can be estimated followed by *directional interpolation* (DI) applied along the estimated edge direction(s). Hsia’s method [14], estimates the edge direction of a lost MB through one-dimensional boundary matching. The methods described in [15–18] estimate the most likely edge orientations in the neighborhood of the missing MB by applying edge detectors to the adjacent blocks/MBs. Linear interpolation along detected edges that pass—when extended—from the missing macroblock is then performed in the methods of [15, 16]. In [19] interpolation is done recursively for bands of missing pixels, using the border pixels from surrounding MBs and already concealed pixels of the recovered MB. The choice of border pixels used for the interpolation is based on a cost estimate which favors pixels that preserve the existing (in the adjacent MBs) edge information across the missing MB. This concept, of sequential error concealment, is also followed in [20, 21]. In the method of Younghoon et al. [21] the missing block is segmented into multiple regions based on the edges that are estimated as crossing the missing block. The intra prediction modes of adjacent intra-blocks (possible only in intra-frames) are used for the purpose of estimating the edge directions of the lost MB. This allows a reduced complexity edge detection to be performed in the neighborhood of the missing block in order to estimate edge strengths. Concealment of the missing pixels is realized through directional interpolation and depends on the location of these pixels within the segmented macroblock. If they are located on an edge then interpolation will be performed along the edge direction. If they are located in a “flat area” then appropriate reference pixels that neighbor the particular segment and which follow the direction of the edges defining the boundary of the segment will be selected for concealing these missing pixels.

In areas of increased spatial activity or texture, directional interpolation can result in the creation of false edges or the emphasis of relatively weak ones. This effect is shown in Figure 9.2b. The human visual system is highly sensitive to edge distortions—including both impairments on existing edges and the creation of false edges [22]. In such cases a smooth approximation for the missing MB can give better subjective (and objective) quality. A switching method can thus be used to decide which interpolation approach is more appropriate for the missing macroblock. The method of [23] relies on edge strength values provided by a previous edge detection step and multiple thresholds in order to switch between the bilinear interpolation of [13], directional interpolation, and the best neighborhood matching method (BNM) of [24]. In [25] Tsekeridou et al. employ a seemingly complex split and match approach for neighboring blocks in order to decide which SEC method is appropriate. In contrast, the method of Agrafiotis et al. [26] calculates the *directional entropy* of detected edge pixels in spatially

**FIGURE 9.2**

Concealment of the missing MBs shown in (a) ("foreman" frame 270) using DI (b) and BI (c).

**FIGURE 9.3**

Concealment of the missing MBs shown in Figures 9.1 and 9.2 with the method of [26].

adjacent correctly received macroblocks in order to select between an edge preserving concealment approach (directional interpolation) or one that gives a smooth approximation of the missing data (bilinear interpolation). Large directional entropy indicates that no specific edge direction is perceptually prevalent and the use of DI in this case would not be suitable as it would lead to the creation of false strong edges. The results presented (Figure 9.3) indicate that one of the advantages of this method is the fact that it does not experience dips in performance compared to single interpolation methods.

In [27] Rane et al. use a coarseness measure to classify the missing blocks as textured or structured. Lost blocks classified as structured are reconstructed using the image inpainting algorithm of [28], while texture synthesis, as described in [29], is used for the textured blocks. The coarseness measure is given by the number of local extrema in the neighborhood of the lost block, where the local extrema are the pixels which are local row extrema as well as local column extrema. A very simple edge detector is

additionally employed to better identify structured blocks. The method of [30] additionally deals with the issue of structure within texture blocks and vice versa by applying structure-aware texture synthesis, a method that combines sparse-modeling and patch-based synthesis. Segmentation is first performed in the neighborhood of the missing blocks in order to identify a plausible set of segments that are related to the area to be filled. The results presented in [30] look very promising with the method being able to reconstruct structure within textured areas even for large unknown blocks.

### 5.09.2.3 Variations

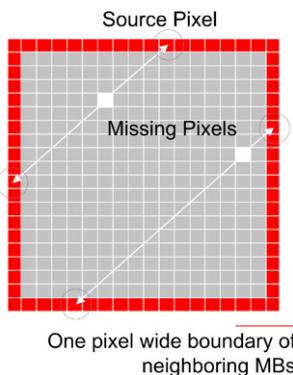
Several variations on the above approach exist. For example, the method of [24] uses a search process in the vicinity of the missing data, in order to replace whole missing blocks with similar ones located in the same frame. A blockwise luminance transformation is used to find the best match for the region surrounding the missing block of pixels within a search region that is located in the same frame as the missing block. The method of [31] estimates missing spatial information in the frequency domain, unlike all previous methods that operate in the spatial domain. Hybrid methods making use of both domains have also been reported in [32, 33].

### 5.09.2.4 Required techniques

#### 5.09.2.4.1 Directional interpolation

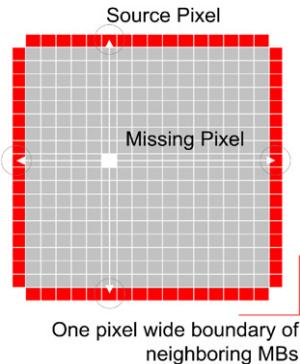
Directional interpolation is a weighted 1D interpolation along a specific edge direction (Figure 9.4). The weights chosen are inversely proportional to the distance between the missing pixel and the source pixels. The source pixels are defined as the two pixels lying on the one-pixel wide external boundary of the missing MB which are intersected by the edge of the specific interpolation direction that passes from the missing pixel. The interpolation is based on the following formula:

$$Mp = \frac{Dsp_2 \times Sp_1 + Dsp_1 \times Sp_2}{Dsp_1 + Dsp_2}, \quad (9.1)$$



**FIGURE 9.4**

Directional interpolation.

**FIGURE 9.5**

Bilinear interpolation.

where  $M_p$  is the missing pixel,  $Sp_1$  and  $Sp_2$  are the two source pixel values along the interpolation direction, and  $D_{Sp_1}$  and  $D_{Sp_2}$  are the distances from the missing pixel to source pixels  $Sp_1$  and  $Sp_2$ , respectively. If the selected direction crosses the external boundary at a non-integral position then the nearest pixel on the boundary is selected.

#### 5.09.2.4.2 Bilinear interpolation

Bilinear interpolation replaces each missing pixel with a weighted average of the nearest pixels on the boundary of the 4-neighboring MBs as shown in Figure 9.5. The weights used are inversely proportional to the distance of source and destination pixels. The interpolation is done according to the following formula:

$$M_p = \frac{D_R \times Sp_L + D_L \times Sp_R + D_T \times Sp_B + D_B \times Sp_T}{D_R + D_L + D_T + D_B}, \quad (9.2)$$

where  $M_p$  is the missing pixel,  $Sp_L$ ,  $Sp_R$ ,  $Sp_T$ ,  $Sp_B$  are the left, right, top, and bottom source pixels, and  $D_L$ ,  $D_R$ ,  $D_T$ , and  $D_B$  are the corresponding distances from the missing pixel.

#### 5.09.2.4.3 Directional entropy

The entropy calculation takes into account only edge-pixels i.e., those pixels that have edge-related information in the neighborhood of the missing macroblock. A large directional activity indicates that no specific edge direction is perceptually prevalent since edges of different directions interact with each other. The directional entropy is given by:

$$H_d = - \sum_{x \in EP} p(d_x) \log_2 p(d_x), \quad (9.3)$$

where  $EP$  is the set of pixels  $x$  in the neighborhood of the missing MB that have associated edge data and  $d_x$  is the directional category of the edge at the specific pixel  $x$ . The maximum possible entropy for 8 directional categories is 3 bits which corresponds to all directions being equally probable. An entropy threshold of 2.6 is used in [26]. Figure 9.3 shows how this entropy changes for the MBs adjacent to

**Table 9.1** Average PSNR Performance for all Concealed IDR Frames Across all Error Patterns

PSNR (dB)	Foreman	Stefan	Bus	Art
BI	30.75	26.25	27.09	28.52
DI	32.49	25.74	27.14	28.63
[23]	32.08	26.34	27.83	28.76
[26]	<b>32.84</b>	<b>26.35</b>	<b>27.97</b>	<b>28.80</b>

those missing in the two corrupted frames of the “foreman” sequence shown in Figures 9.1 and 9.2. Note that MBs with a clear dominant edge direction have a low directional entropy (dark colors) while MBs cluttered with edges have a higher entropy (light colors).

### 5.09.2.5 Performance of SEC methods

Comparing the performance of all available spatial error concealment techniques is very difficult due to the multitude of SEC methods reported in the literature. Herein we present results (first published in [26]) that offer an indication of the relative performance of some of the methods mentioned in the previous sections. The results compare the performance of the multi-interpolation SEC method of [26] with that of the directional interpolation method followed by, among others, [34, 15, 16], the bilinear interpolation method of [35, 13], and a partial implementation of the method in [23] using the same classification thresholds but with the BNM case substituted by BI. Four CIF test sequences were used encoded with H264 (JM) at 1 Mbit/sec with 1 IDR frame every 12 P frames, using dispersed FMO and slice sizes of 66 MBs. Errors in the form of slice erasures were introduced randomly to the coded bitstream at a rate of 4%. Ten different random error patterns were used with each sequence. With each error pattern a different set of IDR frames was affected. The results presented are only for the affected IDR frames in the form of average frame PSNR—i.e., average PSNR of all such spatially concealed IDR frames (Table 9.1). The results show that the multi-interpolation method of [26] performs significantly better compared to BI (up to 2 dB) or a fixed DI approach (up to 0.8 dB) as well as the method of [23]. One of the advantages of the multi-interpolation method of [26] is the fact that it doesn’t experience dips in performance as do the fixed interpolation approaches. Concealed frames are shown in Figure 9.6.

---

## 5.09.3 Temporal error concealment (TEC)

### 5.09.3.1 Ingredients

- One or more candidate replacement motion vectors.
- Selected boundary pixel values or blocks of pixels.
- A boundary matching method.
- A motion estimation method (optional).
- One or more reference frames.
- Enhancements (optional—for improving the final result).

**FIGURE 9.6**

SEC results for frame 240 of the “bus” sequence. Top, left—error free frame PSNR 38.05 dB, right—corrupted 15.94 dB. Middle, left—BI 27.98 dB, right—DI 28.08 dB. Bottom, left—[23] 28.59 dB, right—[26] 29.56 dB.

### 5.09.3.2 Method

A low complexity and popular (implementationwise) approach to temporally concealing a missing macroblock is to replace the missing motion vector(s) of the corrupted MB directly with a candidate replacement motion vector without first evaluating the suitability of that candidate replacement MV [1,4–6,8, 36]. This could be the zero motion vector (which leads to replacement of the corrupted MB by the collocated MB in the previous frame), the MV of an adjacent correctly received MB, or the average (or median) of all such available (or newly estimated as in [37]) adjacent MVs. The end result with this approach is only adequate in cases of very smooth or zero motion, and even then performance is often poor.

### 5.09.3.2.1 Matching measures

Improved results can be obtained by using a matching measure (selection criterion) to evaluate the candidate MVs prior to replacement. The choice of matching measure plays a central role in the success of the concealment process, as shown in the study performed by Agrafiotis et al. in [26]. One possible measure is the sum of absolute differences (SAD) along the outer one-pixel boundary between the recovered MB and those surrounding the lost MB, as suggested by the boundary matching algorithm (BMA) of [38]. The vector that minimizes this boundary matching error is selected to replace the missing MV(s). BMA effectively favors motion vectors that lead to a smooth spatial transition between neighboring macroblocks. Modifications to this basic approach include weighting the border SAD by the variance of pixels lying on the specific boundary [34] or using edge-directed BMA [39] wherein the SAD is calculated along detected edge directions (for each pixel on the external boundary of the restored MB the spatial boundary distortion is calculated along the edge direction). In [40] Bernadini et al. combine BMA with a motion smoothness constraint whereby the value of the replacement MV should be within certain limits compared to the average of the neighboring MVs. The resulting method is reported to compare favorably with existing state-of-the-art methods, both objectively and subjectively.

Significantly better results (up to 3 dB [26] compared to BMA) can be obtained if the external boundary matching error (EBME)—also referred to as outer boundary matching algorithm (OBMA) in the literature—is used as the matching measure [41–43]. In this case, MVs are evaluated in terms of the SAD or MAD of the external boundary pixels for the lost MB and the same external boundary pixels of the candidate replacement MB. EBME favors motion uniformity/smoothness, i.e., the motion of the missing MB should be similar to the motion of one or more of the adjacent MBs. Variations of this matching measure can be obtained by altering the boundary width from 1 to 8 pixels [41–43], using a weighted SAD [44] or resorting to full external block ( $16 \times 8$ ,  $8 \times 16$ , or  $16 \times 16$ ) matching [25]. A 2-pixel wide EBME is recommended as it offers almost all the benefit of any other external boundary matching error, but with low complexity [26].

### 5.09.3.2.2 Selecting motion vector candidates

Having decided on a boundary matching measure, a list of candidate replacement MVs, that will be tested by the chosen matching measure, can be formed. Many methods employ motion estimation (ME) for forming motion vector candidates [19,25,37,40–42,45–47]. This implies a search for the best matching external boundary (assuming that EBME is adopted) in a previous frame and within a specified search range. The complexity of such an approach is however high and, as a result, some methods recommend the use of fast search algorithms [25]. Other methods limit the search range around a starting point based on an existing MV [37]. Hsu et al. in [45] examine the motion of the neighboring correctly received MBs and avoids the use of ME in the case of this motion being zero. Alternatively existing MV candidates can be used (e.g., neighboring MVs, the average or median of those, the MV of past collocated MBs, zero MV, etc.) [34,35,38,41,44,48]. Extrapolation of MVs of collocated and neighboring MBs in the previous frame is suggested in [49]. This adds another MV candidate to the above list in the form of the weighted average of these extrapolated MVs. Combinations of both approaches (use of both ME and neighboring MVs) are suggested in [42,46,47]. Temporal activity thresholds are used in [42,47] for triggering ME and for defining the search range. The list of MV candidates suggested in [26] includes the zero MV, the MV of the 8-neighbors, the MV of the collocated MB, and that of its collocated 8-neighbors.

Macroblocks are typically partitioned into sub-blocks for more accurate motion estimation. The partitioning is usually guided by a rate-distortion process performed during encoding. As a result a missing macroblock can correspond to multiple missing MVs. Any TEC method can estimate more than one MV for each missing MB by applying the matching measure calculation on a sub-block basis as is done in [34, 42]. Such an approach can however lead to block discontinuities among the sub-blocks, requiring post-filtering as described in [42]. Deng [39] utilizes the detected edge directions to identify whether the lost MB contains multiple objects and decide on how to best conceal partitions (sub-blocks) of it. The same goal of partitioning the missing macroblock is targeted by the adaptive error concealment block size selection algorithm of [30]. The motion-compensation modes (i.e., block-partitioning types) of surrounding macroblocks are employed to predict the mode of a lost macroblock which is then concealed on a sub-block or macroblock basis. Multiple MVs can also be estimated by the motion field interpolation methods of [50–56]. Aggregation of MBs rather than partition is the target of [57] where a size-adaptive region boundary matching is employed for concealing missing slices. Instead of using MSE or MAE as the matching criteria, a structural similarity (SSIM) based criterion is employed. The algorithm proposed in [58] by Lie et al. also considers slice errors and uses Dynamic Programming based optimization to estimate the lost MVs in a combined manner that not only considers boundary matching errors, as in BMA, but also side-smoothness between recovered MBs.

### 5.09.3.2.3 Motion vector refinement

Once a replacement MV has been selected, the initial result can be further enhanced. One way of achieving this is via motion refinement of the selected MV. Motion refinement implies that the selected MV is used as a starting point for a motion estimation process that searches for better MV replacements using the chosen matching measure. For complexity reasons, motion refinement should normally be implemented using a fast ME algorithm. The performance enhancements offered by motion refinement were studied by Agrafiotis et al. in [26] for the case of refining the winning (selected) MV, and the zero MV. The performance benefit was found to be insignificant in the case of P frames. In I frames however, which lack motion information and hence good MV candidates, motion refinement proved to be a useful step. More specifically in [26] the replacement MV for missing macroblocks in intra coded frames was selected among the zero MV, the MVs of the collocated MBs, and their 8-neighbors in the previous P frame. Motion refinement of the winning MV led to improvements of up to 1.8 dB. A method to detect motion vector outliers is proposed in [57] as the last stage of their error concealment scheme.

Other enhancements include *overlapped block motion compensation* (OBMC) whereby multiple MVs are used for concealing a missing macroblock [59, 44]. OBMC for concealment implies use of more than one reference MBs (and hence more than one estimated MVs) for the replacement of a damaged macroblock. Differences between methods include the weights used and the matching measure employed for the central replacement MB pixels. OBMC is shown in [26] to offer improvements of around 0.3–0.5 dB for concealing corrupted P frames.

Chen et al. [60] do not directly copy the replacement pixels pointed to by the selected (winning) MV into the location of the missing block but instead instigate a refinement process that aims to reduce blocking artifacts while preserving the structures of the reference MB. The refinement process uses a partial differential equation (PDE) based algorithm to minimize, in a weighted manner, the difference between the gradient field of the reconstructed MB in the current frame and that of the reference MB in the reference frame under a given boundary condition. In [61] the refinement process sees each pixel within

the corrupted block being replenished as the weighted summation of pixels within a square centered at the pixel indicated by the derived motion vector. This is done through the use of an auto-regression algorithm which the authors claim improves both the objective and subjective quality of the replacement blocks.

The order of the steps in the concealment process can affect the end result. For example, in heavily corrupted areas (large/many missing slices) previously concealed macroblocks have to be used as a source of candidate motion vectors and of boundary pixels. In this case, a typical order follows an onion layer approach with outer rings of macroblocks being concealed first. An enhanced edge-sensitive processing order for temporal error concealment is proposed in [62] that modifies the “normal” processing order in order to improve the concealment performance.

### 5.09.3.3 Variations

The method described in [63] by Ma et al. combines spatial concealment elements with temporal concealment. The strong edges in the corrupted frame are first estimated based on the edges in the neighboring frames and the correctly received area of the current frame. The lost regions along these estimated edges are recovered using both spatial and temporal neighboring pixels. The remaining parts of the lost regions are concealed using a purely TEC approach.

Multi-frame TEC approaches have also been reported in [64, 65] which exploit more than one past and/or future frame at the cost of increased complexity and delay. The method in [66] treats the video sequence as a space-time volume (5 past and 5 future frames are employed) and exploits temporal-spatial continuity of video sequences from a global view. Their approach is formulated into a total variation regularized matrix completion model.

### 5.09.3.4 Required techniques

#### 5.09.3.4.1 Boundary matching error

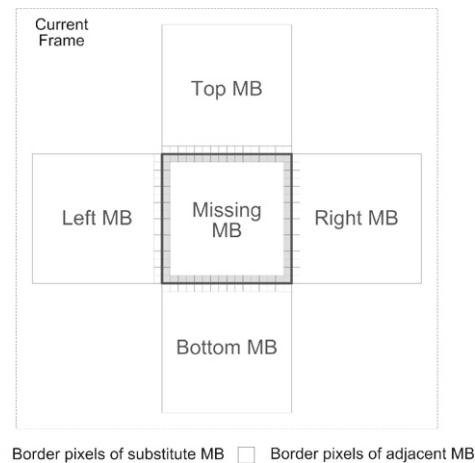
The boundary matching error (BME) (Figure 9.7) is defined as the sum of absolute differences along the one-pixel boundary between the recovered MB and the surrounding ones. It is given by the following formula:

$$\text{BME} = \sum_{x=x_0}^{x_0+N-1} \left[ \left| (F_{x,y_0-1} - F_{x+u_x,y_0+u_y}^r) \right| + \left| (F_{x,y_0+N} - F_{x+u_x,y_0+N-1+u_y}^r) \right| \right] \\ + \sum_{y=y_0}^{y_0+N-1} \left[ \left| (F_{x_0-1,y} - F_{x_0+u_x,y+u_y}^r) \right| + \left| (F_{x_0+N,y} - F_{x_0+N-1+u_x,y+u_y}^r) \right| \right], \quad (9.4)$$

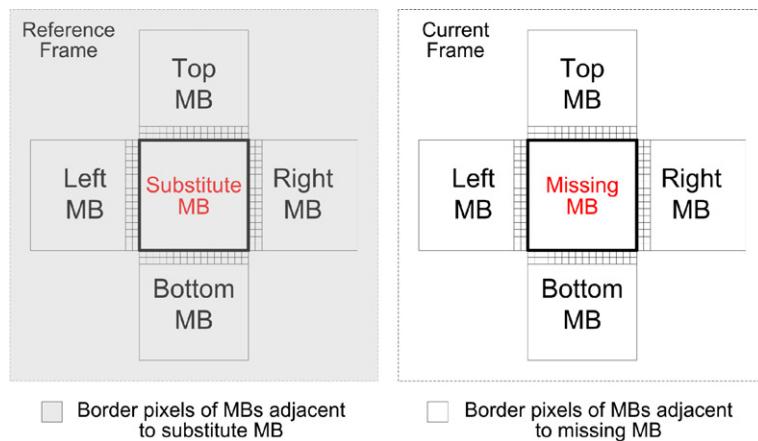
where  $(x_0, y_0)$  is the coordinate of the top-left pixel of the missing MB,  $(u_x, u_y)$  is the candidate MV under consideration,  $N$  is the macroblock size ( $N = 16$ ), and  $F_{x,y}$  and  $F_{x,y}^r$  are the pixels of the current and of the reference frame respectively.

#### 5.09.3.4.2 External boundary matching error

The external boundary matching error (EBME) (Figure 9.8) is defined as the sum of absolute differences between the multiple pixel boundary of MBs adjacent to the missing one in the current frame and the same boundary of MBs adjacent to the replacement MB in the reference frame. It is given by the

**FIGURE 9.7**

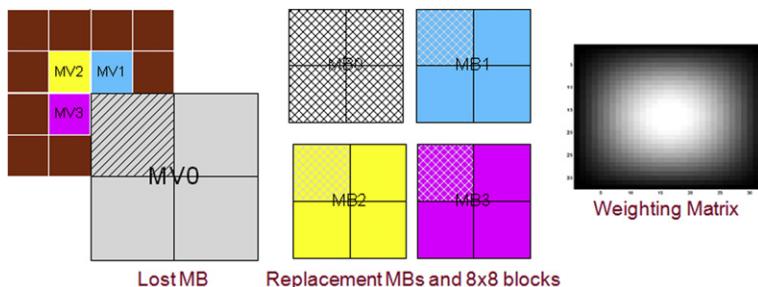
Boundary matching error.

**FIGURE 9.8**

External boundary matching error.

following formula:

$$\begin{aligned}
 \text{EBME} = & \sum_{i=1}^M \sum_{x=x_0}^{x_0+N-1} \left[ \left| \left( F_{x,y_0-i} - F_{x+u_x, y_0-i+u_y}^r \right) \right| + \left| \left( F_{x,y_0+N-1+i} - F_{x+u_x, y_0+N-1+i+u_y}^r \right) \right| \right] \\
 & + \sum_{i=1}^M \sum_{y=y_0}^{y_0+N-1} \left[ \left| \left( F_{x_0-i,y} - F_{x_0-i+u_x, y+u_y}^r \right) \right| + \left| \left( F_{x_0+N-1+i,y} - F_{x_0+N-1+i+u_x, y+u_y}^r \right) \right| \right], \tag{9.5}
 \end{aligned}$$

**FIGURE 9.9**

Overlapped block motion compensation (OBMC).

where  $(x_0, y_0)$  is the coordinate of the top-left pixel of the missing MB,  $(u_x, u_y)$  is the candidate MV under consideration,  $N$  is the macroblock size ( $N = 16$ ),  $M$  is the width of the boundary, and  $F_{x,y}$  and  $F_{x,y}^r$  are the pixels of the current and of the reference frame respectively.

#### **5.09.3.4.3 Overlapped block motion compensation**

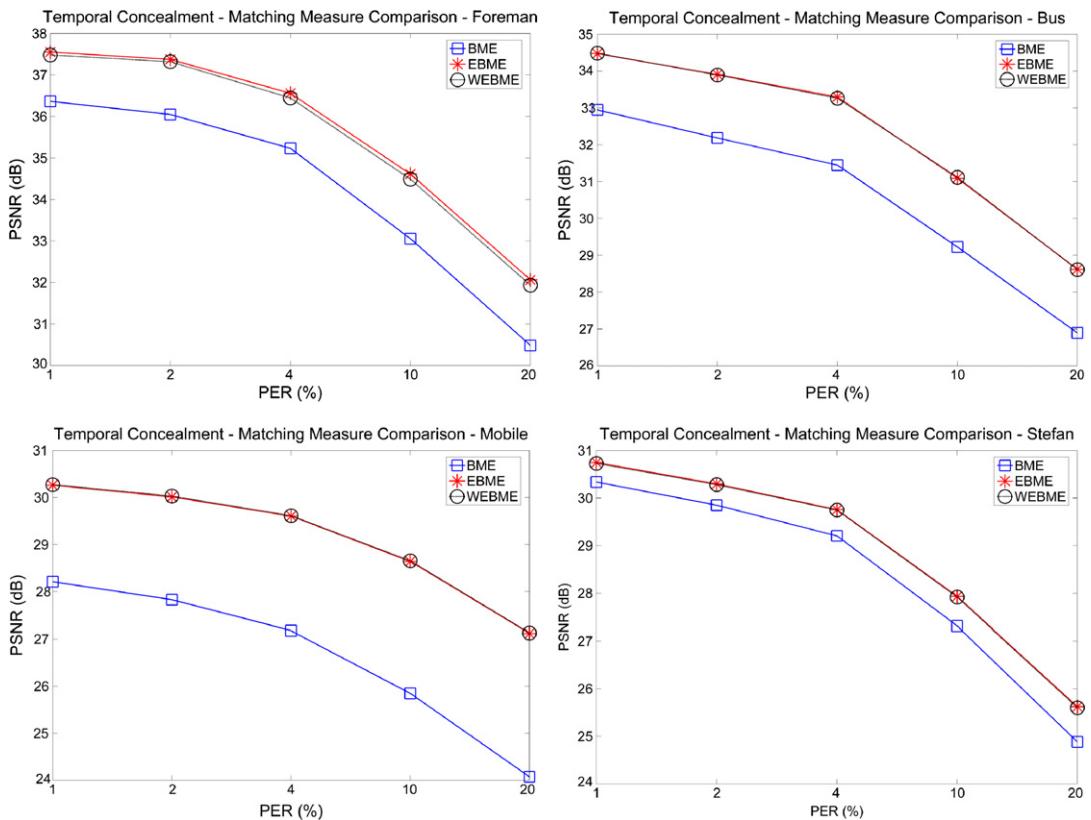
Overlapped block motion compensation (OBMC) for concealment is implemented as follows (Figure 9.9):

1. The damaged MB is divided into four  $8 \times 8$  blocks.
2. Four replacement signals are generated for each  $8 \times 8$  block, using the winning MV and the MV of the corresponding adjacent  $8 \times 8$  blocks of the three neighboring MBs.
3. The four replacement signals are blended according to a raised cosine weighting matrix that favors the winning MB for replacing pixels close to the center of the lost MB.

#### **5.09.3.5 Performance of TEC methods**

As with SEC methods, the multitude of TEC methods reported in the literature renders the task of comparing their performance nearly impossible. However the breakdown of existing methods into the concealment features that make up each algorithm, as described in the previous sections, allows the evaluation of the contribution of each of these concealment features to the performance of the final algorithm (Figures 9.10–9.13). Such an evaluation was performed by Agrafiotis et al. [26] and is repeated herein. Readers may want to use this evaluation when building up a new temporal error concealment method for deciding which of these features they wish to adopt in their method. Selecting the best of these features leads to the formation of the method described in [26] which offers significant improvements relative to the BMA approach adopted by the JM H.264 decoder [35] (Table 9.2).

For the results given below, three H.264/AVC test sequences were coded at 1 Mbit/sec using IDR (1/12) and P frames with a slice size equal to 536 bytes, dispersed FMO with two slice groups, 3–5 reference frames, and a search range of 32 pixels. Results are shown in terms of the average PSNR of the corrupted P frames after concealment, for the cases of the coded bitstream being affected by random packet errors (slice erasures) at rates of 1%, 2%, 4%, 10%, and 20% (results are averaged over 10 error

**FIGURE 9.10**

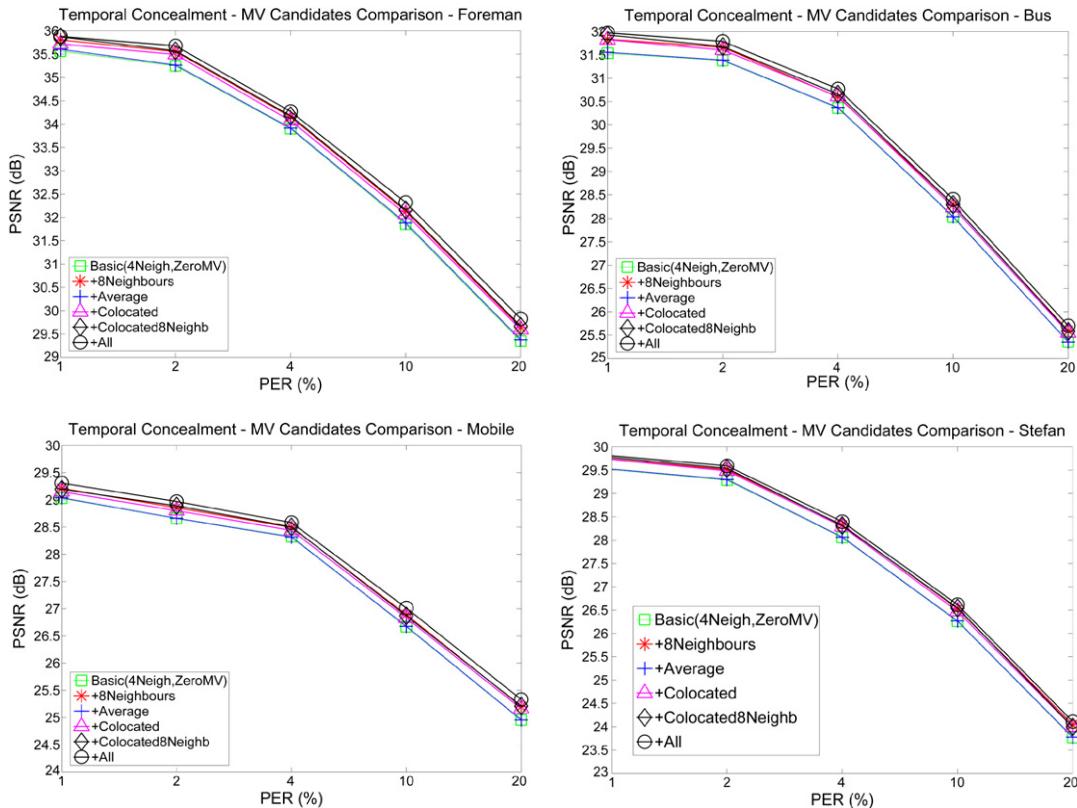
Performance comparison of matching measures for temporal concealment. Average PSNR of corrupted and subsequently concealed P frames is shown for sequences "foreman," "bus," "mobile", and "stefan," for different packet error rates (PER). The graphs indicate that using an external boundary matching measure leads to better concealment of errors.

patterns). Only P frames were corrupted. For EBME, two different implementations were considered: one using a 2-pixel wide boundary and one using an 8-pixel wide boundary with a raised cosine matrix determining the influence of the boundary pixels on the matching error calculation (WEBME).

## 5.09.4 Mode selection

### 5.09.4.1 Ingredients

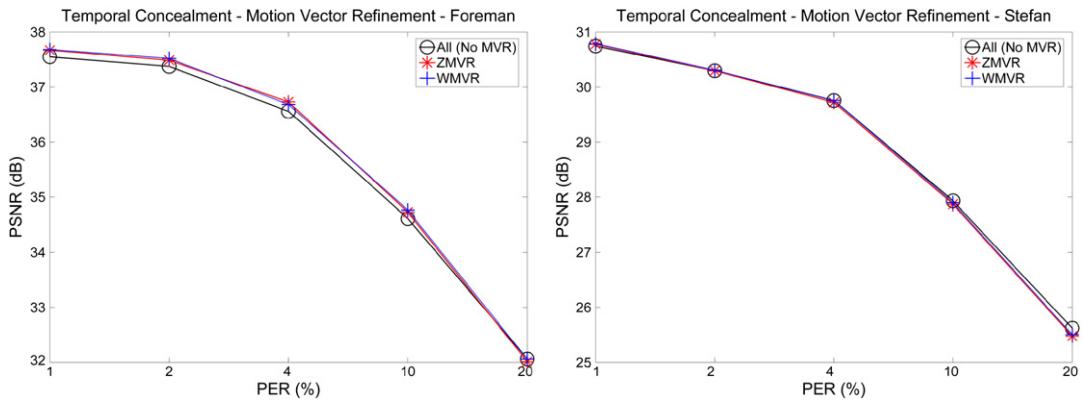
- Knowledge of coding modes for neighboring blocks (optional).
- Selected boundary pixel values or block of pixels.
- One or more reference frames.

**FIGURE 9.11**

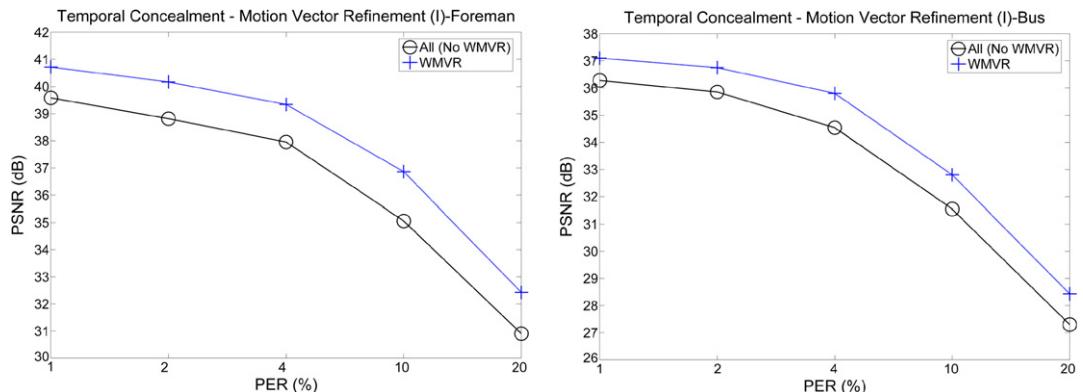
Performance comparison of motion vector candidates for temporal concealment. Average PSNR of corrupted and subsequently concealed P frames is shown for sequences “foreman,” “bus,” “mobile”, and “stefan” for different packet error rates (PER). The graphs show the enhancement in PSNR performance offered by each additional MV candidate.

### 5.09.4.2 Method

Having identified appropriate spatial and temporal concealment methods, it is then necessary to decide when to deploy them, i.e., which method to use for every missing macroblock. This decision can be fixed a priori using spatial concealment for intra coded frames (I) and temporal concealment for predictive frames (P, B) as is reported in [35]. Both [35, 25] effectively assume continuous high correlation between the frames of the coded sequence thus suggesting the use of TEC methods in all P frames. This however is not always the case, both at the frame level (scene changes) and at the macroblock level (objects appearing/disappearing). More complex mode decision algorithms are described in [36, 34, 67–70]. Sun et al. [36] use measures of temporal and spatial activity to decide which concealment mode will be used in I frames for replacing the missing MB. In [34, 67, 68, 70] the coding modes of the neighboring MBs

**FIGURE 9.12**

TEC results for corrupted P frames using the previously tested parameters (All—No MVR) and with additional motion refinement of the zero MV (ZMVR) and the winning MV (WMVR).

**FIGURE 9.13**

TEC results for corrupted I frames using the previously formulated parameters (All—No WMVR) and with additional motion refinement of the winning MV (WMVR).

determine the type of concealment in P frames; if the majority of these MBs are coded in INTRA mode then the missing MB is concealed spatially, otherwise TEC is applied. In [70] the boundary matching error of the temporally recovered MBs is further compared with that of surrounding MBs. If it is above a specific level then a mismatch is detected and spatial error concealment is applied.

Clearly the coding mode based algorithm cannot be used in I frames and Genari [67] suggests a mode decision algorithm specifically for I frames, wherein motion estimation is applied to 4 MBs and the SAD between predicted and reference MBs is used as an indication of good or bad correlation with the previous frame. In [68] Su et al. perform I frame mode decision based on MB type matching between

**Table 9.2** Average PSNR Performance for all Concealed P Frames Across Ten Error Patterns

PSNR (dB)	PER 1 (%)		PER 4 (%)		PER 10 (%)	
	BMA	[26]	BMA	[26]	BMA	[26]
Mobile	28.51	<b>30.43</b>	27.54	<b>29.80</b>	26.19	<b>28.89</b>
Bus	33.21	<b>34.88</b>	31.72	<b>33.77</b>	29.42	<b>31.61</b>
Stefan	30.46	<b>31.07</b>	29.36	<b>30.15</b>	27.43	<b>28.38</b>
Silent	38.66	<b>39.41</b>	37.38	<b>38.32</b>	35.78	<b>36.88</b>
Foreman	36.60	<b>37.89</b>	35.44	<b>36.97</b>	33.26	<b>35.07</b>
Football	28.14	<b>28.94</b>	27.21	<b>28.09</b>	25.10	<b>26.15</b>
Hall	38.16	<b>39.14</b>	37.14	<b>38.37</b>	35.51	<b>36.91</b>
Tempete	31.81	<b>32.46</b>	31.14	<b>31.97</b>	29.75	<b>30.87</b>
Container	39.47	<b>40.52</b>	39.01	<b>40.31</b>	37.78	<b>39.58</b>
Fl. garden	29.49	<b>31.08</b>	28.65	<b>30.62</b>	26.83	<b>29.24</b>

the temporally adjacent P frames (i.e., previous and next), which adds delay to the system. The method of [69] proposes a gradient boundary matching error for mode selection that penalizes directional edge discontinuity. Can et al. [71] advocate the use of decision trees for selecting a concealment mode.

The concealment mode selection algorithm described in [26] examines the suitability of the TEC method for concealing each missing MB, by evaluating the levels of *motion compensated temporal activity* (MCTA) in the neighborhood of that MB and comparing them with *spatial activity* (SA). TEC is applied if temporal activity is less than the spatial activity or less than a threshold. If TEC is found unsuitable for a specific macroblock, SEC is used for concealing the particular MB. Significant gains in performance (up to 4.6 dB) are reported relative to other concealment mode selection strategies (Figure 9.14). Agrafiotis et al. [26] also describe a variation of this method (MCTA+) that additionally considers the coding modes of the neighboring MBs. More specifically, if the majority of these MBs are intra coded a break in correlation for the specific missing MB is assumed and the algorithm switches to spatial concealment. Otherwise concealment proceeds using the MCTA mode selection algorithm. In I frames the mode selection process becomes equivalent to MCTA since useful coding mode information is not available (all MBs are coded in intra mode). MCTA+ offers additional (small) improvements in concealment performance.

#### 5.09.4.3 Variations

In [72] a sequential error concealment algorithm is presented based on sparse linear prediction that automatically adapts itself to SEC, TEC, or a combined SEC/TEC scheme according to the available information. The method inherently examines the amount of spatial and temporal correlation in the vicinity of a missing macroblock and conceals the missing data using purely spatial, temporal, or both types of neighboring pixels.



**FIGURE 9.14**

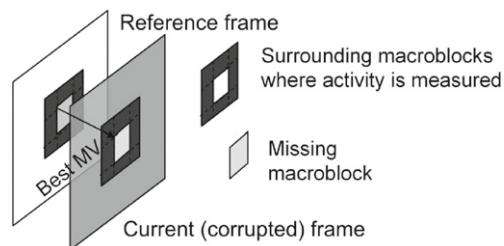
Mode selection results with the method of [26] for P frame 131 (scene change) of “tennis.” Top, left—error free, top right—corrupted (propagated and new errors). Bottom, left—TEC, bottom right—mode selection.

#### 5.09.4.4 Required techniques

##### 5.09.4.4.1 Measures of temporal and spatial activity: MCTA and SA

Motion compensated temporal activity (MCTA) is measured as the mean squared error between the MBs surrounding the missing one in the current frame and those surrounding the replacement MB in the reference frame (Figure 9.15):

$$\text{MCTA} = E[(x - x^*)^2], \quad (9.6)$$



**FIGURE 9.15**

Activity measures for mode selection.

where  $x$  are the pixels in the neighborhood of the missing MB and  $x^*$  are the pixels in the neighborhood of the replacement MB in the reference frame. Spatial activity is measured as the variance of the surrounding MBs in the current frame and is given by:

$$SA = E[(x - \mu)^2]. \quad (9.7)$$

#### 5.09.4.5 Performance of mode selection methods

To compare the performance of mode selection algorithms, TEC and SEC have to be fixed in order to isolate the effect of mode selection on the concealment performance. This was the approach followed in the comparison performed by Agrafiotis et al. in [26], wherein TEC and SEC were set to the methods described in [26]. The mode selection methods of [34, 36, 26] were compared alongside the basic mode selection approaches used in [35] (TEC for P frames SEC for I frames) and [25] (TEC for all frames apart from the 1st Intra-frame). An extension of the method of [36], that can handle both I and P frames, was also tested (referred to as ExtMS[36]).

**Table 9.3** Average Sequence PSNR Across Ten Error Patterns with Different Mode Selection (Fixed TEC and SEC)

Seq.	PER (%)	MS [35]	MS [25]	MS [34]	MS [36]	ExtMS [36]	MCTA [26]	MCTA+ [26]
Anim5	2	39.44	39.76	40.34	39.82	40.72	40.76	<b>40.90</b>
	4	37.14	37.57	38.46	37.67	39.04	39.08	<b>39.35</b>
	10	32.66	32.73	34.70	33.05	35.21	35.48	<b>35.82</b>
Anim6	2	37.87	39.09	38.34	39.10	39.65	39.66	<b>39.73</b>
	4	35.39	36.97	36.17	37.03	38.01	38.06	<b>38.22</b>
	10	30.97	32.78	32.05	33.00	34.50	34.62	<b>34.87</b>
Anim8	2	35.95	35.98	36.41	36.13	36.70	36.75	<b>36.79</b>
	4	34.27	34.22	35.16	34.59	35.60	35.73	<b>35.79</b>
	10	30.86	30.84	32.06	31.29	32.67	32.94	<b>33.06</b>
Tennis	2	35.72	<b>38.16</b>	35.73	37.91	37.26	38.11	38.11
	4	33.23	<b>36.92</b>	33.24	36.60	35.71	36.90	<b>36.92</b>
	10	29.56	34.08	29.58	33.67	32.37	<b>34.19</b>	34.18
Foreman	2	37.75	38.69	37.77	38.63	38.40	38.74	<b>38.75</b>
	4	36.31	37.72	36.32	37.63	37.17	37.80	<b>37.81</b>
	10	32.55	34.63	32.56	34.55	33.83	34.77	<b>34.79</b>
Stefan	2	31.10	32.16	31.11	31.70	30.66	<b>32.20</b>	<b>32.20</b>
	4	29.29	30.83	29.31	30.21	28.68	<b>30.94</b>	<b>30.94</b>
	10	25.97	27.64	25.98	26.93	24.79	27.86	<b>27.87</b>
Bus	2	34.00	<b>35.45</b>	34.00	34.95	33.91	35.43	35.43
	4	32.07	<b>34.04</b>	32.08	33.32	31.73	34.02	34.03
	10	28.33	30.72	28.34	29.86	27.61	30.83	<b>30.84</b>

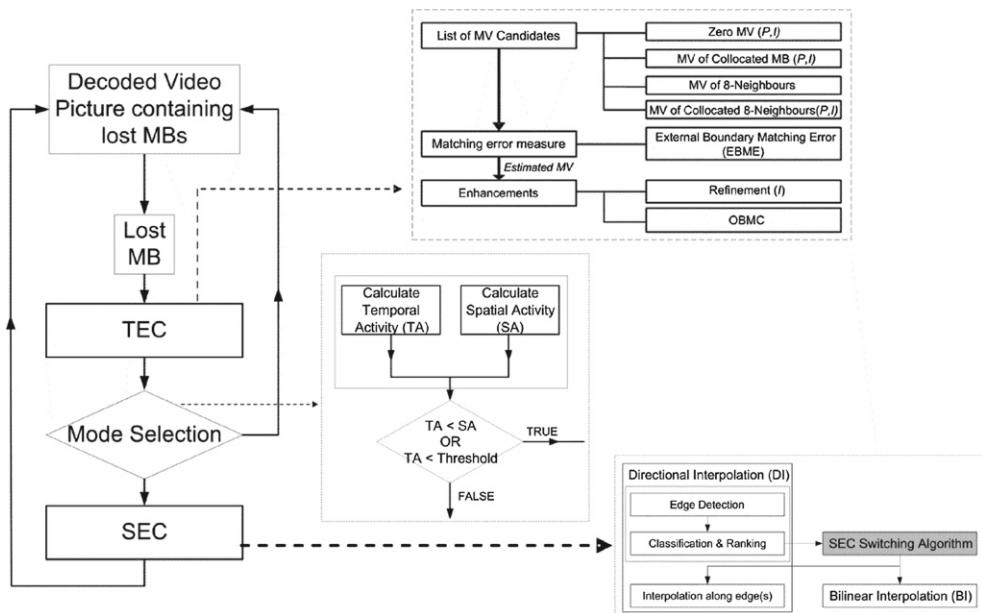
Test sequences were coded with similar parameters as in [Section 5.09.3.5](#) (performance of TEC methods): 1 Mbit/s, IDR (1/12) and P frames, slice size of 536 bytes, dispersed FMO with 2 slice groups, 3–5 reference frames, search range of 32 pixels, with random packet errors affecting both P and I frames this time. Both natural test sequences (“tennis,” “foreman,” “Stefan,” “bus”) and animation sequences (“anim5,” “anim6,” “anim8”) were used. The latter represent a significant challenge for the concealment mode selection process due to the frequent scene changes, multiple objects appearing or disappearing, and very high and irregular motion. Results were averaged across 10 error patterns.

[Table 9.3](#) shows the results of the performance comparison performed in [26]. MCTA and MCTA+ outperformed all other mode selection methods, in some cases significantly so; at a PER of 10% there is a 4.6 dB gain over MS[34] and MS[35] with the “Tennis” sequence, 3.2 dB over ExtMS[36] with “Bus,” 2.77 dB and 3.1 dB over MS[36] and MS[25] respectively with “Anim5.” Even the few times that a fully temporal concealment approach (MS[25]) works better, it does so by a trivial margin (less than 0.05 dB). Although the MCTA algorithm seems to perform slightly worse than MCTA+, its use could prove more suitable for cases where randomly selected slices or MBs are coded in intra mode for improved error resilience. In such cases, mode selection based on coding mode, as done in MCTA+, would not be as reliable.

## 5.09.5 Discussion and conclusions

All the error concealment methods described in the literature and included in this chapter, report some degree of performance gain. This is usually stated relative to competing method(s) that are relatively easy to implement (e.g., TEC using the average MV, or SEC relying on bilinear interpolation alone), or methods that are already available (e.g., the method implemented in the JM H.264 reference software [35]). Often the method described will incorporate a number of concealment features, making it unclear where the performance gain exactly stems from and what the contribution of certain concealment features is. Deciding on which algorithm to use can appear to be a difficult task.

Multi-mode approaches, such as that suggested in [26] by Agrafiotis et al. (and depicted in [Figure 9.16](#)), offer significant performance benefits (up to 9 dB gain relative to the JM H.264 reference software is reported—see [Tables 9.4](#) and [9.5](#)). Extending these to include even more options/sub-algorithms, triggered by local analysis, promises further improvements in performance [73]. Structurally aware texture synthesis [30] is also a strong candidate for spatial error concealment and a comparison with methods that switch between multiple interpolations modes has significant potential. Combining concealment features from the methods reviewed in this paper in a multi-mode switching algorithm could potentially lead to a “super concealment” approach. Including measures of complexity in the decision mechanism, like packet error rates, number of missing blocks, and battery life, would be necessary in order to keep processing time and power consumption under control. Even then however, it is worth bearing in mind that error concealment generally relies on the application of some form of error resilience during encoding (e.g., use of multiple slices, FMO, etc.) in order to stand a chance of moderating the effect of transmission errors. Under a very high rate of bursty errors no concealment strategy would have much success in providing a video of good quality without sufficient amounts of redundancy left/added in the received stream.

**FIGURE 9.16**

The multi-mode error concealment approach of [26].

**Table 9.4** Average Sequence PSNR Across 100 Error Patterns with JM Concealment and the Method of [26]—FMO

PSNR (dB)	EF	PER 1 (%)		PER 4 (%)		PER 10 (%)		PER 20 (%)	
		JM	EECMS	JM	EECMS	JM	EECMS	JM	EECMS
Anim5	43.00	40.75	<b>41.88</b>	36.33	<b>39.31</b>	31.96	<b>35.86</b>	28.14	<b>31.74</b>
Anim6	41.51	39.12	<b>40.56</b>	34.66	<b>38.24</b>	30.22	<b>34.77</b>	26.16	<b>30.38</b>
Tennis	39.41	36.91	<b>38.74</b>	32.39	<b>36.90</b>	28.49	<b>34.11</b>	25.22	<b>30.42</b>
Foreman	40.14	38.11	<b>39.44</b>	34.47	<b>37.65</b>	30.61	<b>34.74</b>	27.33	<b>31.10</b>
Stefan	33.91	32.31	<b>33.02</b>	29.12	<b>30.92</b>	25.68	<b>27.99</b>	22.58	<b>24.74</b>
Mobile	31.36	29.93	<b>31.05</b>	27.12	<b>30.10</b>	23.88	<b>28.30</b>	21.07	<b>25.62</b>
Bus	37.20	35.08	<b>36.33</b>	31.06	<b>34.16</b>	27.17	<b>30.85</b>	23.79	<b>26.99</b>
Football	31.27	30.14	<b>30.56</b>	27.66	<b>28.73</b>	24.83	<b>26.16</b>	22.05	<b>23.22</b>
Tempete	33.87	32.84	<b>33.54</b>	30.45	<b>32.57</b>	27.51	<b>30.81</b>	24.67	<b>28.24</b>
Silent	41.53	39.68	<b>41.00</b>	36.09	<b>39.47</b>	32.49	<b>37.04</b>	29.16	<b>33.77</b>
Hall	40.68	38.73	<b>40.20</b>	34.97	<b>38.77</b>	31.20	<b>36.33</b>	27.64	<b>32.40</b>
Fl. garden	32.12	30.74	<b>31.73</b>	27.76	<b>30.59</b>	24.51	<b>28.53</b>	21.55	<b>25.74</b>
Container	41.42	39.09	<b>41.20</b>	34.91	<b>40.53</b>	30.77	<b>39.15</b>	27.33	<b>36.56</b>

**Table 9.5** Average Sequence PSNR Across 100 Error Patterns with JM Concealment and Method of [26]—NO FMO

PSNR (dB)	EF	PER 1 (%)		PER 4 (%)		PER 10 (%)		PER 20 (%)	
		Concealment	JM	EECMS	JM	EECMS	JM	EECMS	JM
Anim5	43.45	40.39	<b>41.21</b>	34.89	<b>36.74</b>	29.66	<b>31.95</b>	25.66	<b>27.70</b>
Anim6	41.99	38.65	<b>39.97</b>	32.42	<b>35.70</b>	26.91	<b>30.64</b>	22.59	<b>26.10</b>
Tennis	39.76	36.44	<b>38.37</b>	31.04	<b>35.21</b>	26.83	<b>31.37</b>	23.71	<b>27.63</b>
Foreman	40.47	37.80	<b>39.08</b>	33.06	<b>36.02</b>	28.88	<b>32.20</b>	25.37	<b>28.24</b>
Stefan	34.30	31.90	<b>32.90</b>	27.66	<b>29.90</b>	23.79	<b>26.26</b>	20.75	<b>22.82</b>
Mobile	31.74	29.88	<b>31.13</b>	26.39	<b>29.46</b>	22.95	<b>26.79</b>	19.97	<b>23.49</b>
Bus	37.64	34.58	<b>35.92</b>	29.52	<b>32.30</b>	25.02	<b>27.93</b>	21.61	<b>23.91</b>
Football	31.65	30.01	<b>30.40</b>	26.92	<b>27.72</b>	23.68	<b>24.56</b>	20.89	<b>21.57</b>
Tempete	34.21	32.64	<b>33.60</b>	29.39	<b>31.96</b>	26.11	<b>29.50</b>	23.08	<b>26.36</b>
Silent	41.81	39.23	<b>40.85</b>	34.88	<b>38.42</b>	30.55	<b>35.01</b>	26.89	<b>31.43</b>
Hall	40.80	38.44	<b>39.55</b>	33.87	<b>36.69</b>	29.48	<b>32.80</b>	25.80	<b>28.66</b>
Fl. garden	32.47	30.47	<b>31.86</b>	26.88	<b>30.30</b>	23.27	<b>27.60</b>	20.36	<b>24.30</b>
Container	41.68	38.78	<b>41.25</b>	33.56	<b>39.93</b>	28.93	<b>37.50</b>	25.31	<b>33.97</b>

## Glossary

Spatial Error Concealment	the process of replacing missing pixels through the use of spatially adjacent (intra-frame) data
Temporal Error Concealment	the process of replacing missing pixels through the use of temporally adjacent (inter-frame) data
Concealment Mode Selection	the process of deciding which concealment method to use, spatial or temporal
Flexible Macroblock Ordering	error resilience tool supported by H.264 that allows grouping of macroblocks according to certain patterns

## References

- [1] P. Ferré, A. Doufexi, A.R. Nix, D.R. Bull, Packetisation strategies for enhanced video transmission over wireless LANs, in: Proceedings of the Packet Video, 2004.
- [2] D. Agrafiotis, D.R. Bull, T.K. Chiew, P. Ferré, A.R. Nix, Enhanced error concealment for video transmission over wireless LANs, in: Proceedings of International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), April, 2005.
- [3] D. Agrafiotis, T.K. Chiew, P. Ferré, D.R. Bull, A.R. Nix, A. Doufexi, J. Chung-How, D. Nicholson, Seamless wireless networking for video surveillance applications, in: SPIE Proceedings of Image and Video Communications and Processing, January, 2005.

- [4] Y. Wang, S. Wenger, J. Wen, A.K. Katsaggelos, Error resilient video coding techniques-real time video communications over unreliable networks, *IEEE Signal Process. Mag.* 17 (2000) 61–82.
- [5] Y. Wang, Q.-F. Zhu, Error control and concealment for video communication: a review, *Proc. IEEE* 86 (1998) 974–997.
- [6] S. Wenger, H.264/AVC Over IP, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003).
- [7] D. Bull, *Communicating Pictures*, Elsevier, 2014.
- [8] G.J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Trans. Circ. Syst. Video Technol.* 22 (12) (2012) 1649–1668.
- [9] Y. Guo, Y. Chen, Y.-K. Wang, H. Li, M.M. Hannuksela, M. Gabbouj, Error resilient coding and error concealment in scalable video coding, *IEEE Trans. Circ. Syst. Video Technol.* 19 (6) (2009) 781–795.
- [10] B.W. Micallef, C.J. Debono, R.A. Farrugia, Performance of enhanced error concealment techniques in multi-view video coding systems, in: International Conference on Systems, Signals and Image Processing (IWSSIP), June, 2011.
- [11] S. Belfiore, M. Grangetto, E. Magli, G. Olmo, Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation, *IEEE Trans. Multimedia* 7 (2) (2005) 316–329.
- [12] P. Baccichet, D. Bagni, A. Chimienti, L. Pezzoni, F. Rovati, Frame concealment for H.264/AVC decoders, *IEEE Trans. Consum. Electron.* 51 (1) (2005) 227–233.
- [13] P. Salama, N.B. Shroff, E.J. Delp, Error concealment in encoded video streams, in: A.K. Katsaggelos, N.P. Galatsanos (Eds.), *Signal Recovery Techniques for Image and Video Compression and Transmission*, Kluwer Academic Publishers., 1998, (Chapter 7).
- [14] S.-C. Hsia, An edge-oriented spatial interpolation for consecutive block error concealment, *IEEE Signal Process. Lett.* 11 (6) (2004) 577–580.
- [15] J.-W. Suh, Y.-S. Ho, Error concealment based on directional interpolation, *IEEE Trans. Consum. Electron.* 43 (3) (1997) 295–302.
- [16] W. Kwok, H. Sun, Multi-directional interpolation for spatial error concealment, *IEEE Trans. Consum. Electron.* 39 (3) (1993) 455–460.
- [17] H. Sun, W. Kwok, Concealment of damaged block transform coded images using projection onto convex set, *IEEE Trans. Image Process.* 4 (1995) 470–477.
- [18] J. Park, D.-C. Park, R.J. Marks, M.A. El-Sharkawi, Block loss recovery in DCT image encoding using POCS, in: Proceedings of International Symposium on Circuits and Systems (ISCAS), May, 2002.
- [19] M.C. Hong, H. Scwab, L. Kondi, A.K. Katsaggelos, Error concealment algorithms for compressed video, *Signal Process. Image Commun.* 14 (1999) 473–492.
- [20] X. Li, M.T. Orchard, Novel sequential error concealment techniques using orientation adaptive interpolation, *IEEE Trans. Circ. Syst. Video Technol.* 12 (10) (2002) 857–864.
- [21] K. Younghoon, L. Hoonjae, S. Sull, Spatial error concealment for H.264 using sequential directional interpolation, *IEEE Trans. Consum. Electron.* 54 (4) (2008) 1811–1818.
- [22] E. Ong, W. Lin, Z. Lu, S. Yao, M. Etoh, Visual distortion assessment with emphasis on spatially transitional regions, *IEEE Trans. Circ. Syst. Video Technol.* 14 (4) (2004) 559–566.
- [23] Z. Rongfu, Z. Yuanhua, H. Xiaodong, Content-adaptive spatial error concealment for video communication, *IEEE Trans. Consum. Electron.* 50 (1) (2004) 335–341.
- [24] Z. Wang, Y. Yu, D. Zhang, Best neighborhood matching: an information loss restoration technique for block-based image coding systems, *IEEE Trans. Image Process.* 7 (7) (1998) 1056–1061.
- [25] S. Tsekridou, I. Pitas, MPEG-2 error concealment based on block-matching principles, *IEEE Trans. Circ. Syst. Video Technol.* 10 (4) (2000) 646–658.
- [26] D. Agrafiotis, D.R. Bull, N. Canagarajah, Enhanced error concealment with mode selection, *IEEE Trans. Circ. Syst. Video Technol.* 16 (8) (2006) 960–973.

- [27] S.D. Rane, G. Sapiro, M. Bertalmio, Structure and texture filling in of missing image blocks in wireless transmission and compression applications, *IEEE Trans. Image Process.* 12 (3) (2003) 296–303.
- [28] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: *Computer Graphics (SIGGRAPH 2000)*, July 2000, pp. 417–424.
- [29] A.A. Efros, T.K. Leung, Texture synthesis by nonparametric sampling, in: *IEEE International Conference on Computer Vision*, Corfu, Greece, September 1999, pp. 1033–1038.
- [30] H. Lakshman, P. Ndjiki-Nya, M. Koppel, D. Doshkov, T. Wiegand, An automatic structure-aware image extrapolation applied to error concealment, in: *IEEE International Conference on Image Processing (ICIP)*, November 2009.
- [31] K. Meisinger, A. Kaup, Spatial error concealment of corrupted image data using frequency selective extrapolation, in: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.
- [32] Y. Wang, Q.-F. Zhu, L. Shaw, Maximally smooth image recovery in transform coding, *IEEE Trans. Commun.* 41 (10) (1993) 1544–1551.
- [33] Z. Alkachouh, M.G. Bellanger, Fast DCT based spatial domain interpolation of blocks in images, *IEEE Trans. Image Process.* 9 (4) (2000) 729–732.
- [34] Y. Xu, Y. Zhou, H.264 video communication based refined error concealment schemes, *IEEE Trans. Consum. Electron.* 50 (4) (2004) 1135–1141.
- [35] Y.-K. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta, M. Gabbouj, The error concealment feature in the H.26L test model, in: *Proceedings of International Conference on Image Processing (ICIP)*, 2002.
- [36] H. Sun, J.W. Zdepski, W. Kwok, D. Raychaudhuri, Error concealment algorithms for robust decoding of MPEG compressed video, *Signal Process.: Image Commun.* 10 (1997) 249–268.
- [37] J.-W. Suh, Y.-S. Ho, Error concealment techniques for digital TV, *IEEE Trans. Broadcast.* 48 (4) (2004) 299–306.
- [38] W.-M. Lam, A.R. Reibman, Recovery of lost or erroneously received motion vectors, in: *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1993, pp. V-417–V-420.
- [39] X. Deng, Y. Liu, C. Hong, J. Bu, C. Chen, A temporal error concealment algorithm for H.264/AVC based on edge directions, in: *IEEE International Conference on Image Processing (ICIP)*, November 2009.
- [40] R. Bernardini, L. Celetto, G. Gennari, M. Cargnelutti, R. Rinaldo, Error concealment of H.264/AVC inter-coded video frames, in: *International Conference on Image Processing (ICIP)*, September 2010.
- [41] J. Zhang, J.F. Arnold, M.R. Frater, A Cell-loss concealment technique for MPEG-2 coded video, *IEEE Trans. Circ. Syst. Video Technol.* 10 (4) (2000) 659–665.
- [42] T. Chen, Refined boundary matching algorithm for temporal error concealment, in: *Proceedings of Packet Video*, 2002.
- [43] T.S. Valente, C. Dufour, F. Groliere, D. Snook, An efficient error concealment implementation for MPEG4 video streams, *IEEE Trans. Consum. Electron.* 47 (3) (2001).
- [44] Y. Kuo, S.-C. Tsao, Error concealment based on overlapping, in: *Proceedings of VCIP*, San Jose, CA, USA, 2002.
- [45] C.-T. Hsu, M.-J. Chen, W.-W. Liao, S.-Y. Lo, High-performance spatial and temporal error-concealment algorithms for block-based video coding techniques, *ETRI J.* 27 (1) (2005) 53–63.
- [46] B. Yan, K.W. Ng, A novel selective motion vector matching algorithm for error concealment in MPEG-4 video transmission over error-prone channels, *IEEE Trans. Consum. Electron.* 49 (4) (2003) 1416–1423.
- [47] L.-W. Kang, J.-J. Leou, A hybrid error concealment scheme for MPEG-2 video transmission based on best neighborhood matching algorithm, *J. Vis. Commun. Image Represent.* 16 (3) (2005) 288–310.
- [48] G. Sullivan, T. Wiegand, K.-P. Lim, Joint model reference encoding methods and decoding concealment methods, in: *Document JVT-I049*, San Diego, USA, September 2003.

- [49] J.-Y. Pyun, J.-S. Lee, J.-W. Jeong, J.-H. Jeong, S.-J. Ko, Robust error concealment for visual communications in burst-packet-loss networks, *IEEE Trans. Consum. Electron.* 49 (4) (2003) 1013–1019.
- [50] S. Tsekeridou, F. Alaya Cheikh, M. Gabbouj, I. Pitas, Vector rational interpolation schemes for erroneous motion field estimation applied to MPEG-2 error concealment, *IEEE Trans. Multimedia* 6 (6) (2004) 876–885.
- [51] M.E. Al-Mualla, N. Canagarajah, D.R. Bull, Temporal error concealment using motion field interpolation, *IEE Electron. Lett.* 35 (1999) 215–217.
- [52] M.E. Al-Mualla, N. Canagarajah, D.R. Bull, Error concealment using motion field interpolation, in: Proceedings of IEEE International Conference on Image Processing (ICIP), 1998.
- [53] J. Zheng, L.-P. Chau, Error-concealment algorithm for H.26L using first-order plane estimation, *IEEE Trans. Multimedia* 6 (6) (2004) 801–805.
- [54] C. Chen, M. Chen, C. Huang, S. Sun, Motion vector based error concealment algorithms, in: Proceedings of the Third IEEE Pacific Rim Conference on Multimedia, Lecture Notes In Computer Science, vol. 2532, 2002, pp. 425–433.
- [55] J. Zheng, L.-P. Chau, Efficient motion vector recovery algorithm for H.264 based on a polynomial model, *IEEE Trans. Multimedia* 7 (3) (2005) 507–513.
- [56] J. Zheng, L.-P. Chau, A motion vector recovery algorithm for digital video using lagrange interpolation, *IEEE Trans. Broadcast.* 49 (4) (2003) 383–389.
- [57] H. Gao, J.Y. Tham, W.S. Lee, K.H. Goh, Slice error concealment based on size-adaptive SSIM matching and motion vector outlier rejection, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May, 2011.
- [58] W.-N. Lie, C.H. Yeh, Z.W. Gao, Video error concealment by using iterative dynamic-programming optimization, in: International Conference on Image Processing (ICIP), September, 2010.
- [59] M.-J. Chen, L.-G. Chen, R.-M. Weng, Error concealment of lost motion vectors with overlapped motion compensation, *IEEE Trans. Circ. Syst. Video Technol.* 7 (1997).
- [60] Y. Chen, Y. Hu, O.C. Au, L. Houqiang, C.W. Chen, Video error concealment using spatio-temporal boundary matching and partial differential equation, *IEEE Trans. Multimedia* 10 (1) (2008) 2–15.
- [61] Y. Zhang, X. Xiang, D. Zhao, S. Ma, W. Gao, Packet video error concealment with auto regressive model, *IEEE Trans. Circ. Syst. Video Technol.* 22 (1) (2012) 12–27.
- [62] T.-H. Wu, G.L. Wu, C.Y. Chen, S.Y. Chien, Enhanced temporal error concealment algorithm with edge-sensitive processing order, in: International Symposium on Circuits and Systems (ISCAS), May, 2008.
- [63] M. Ma, O.C. Au, S.H.G. Chan, M.T. Sun, Edge-directed error concealment, *IEEE Trans. Circ. Syst. Video Technol.* 20 (3) (2010) 382–395.
- [64] Y.-C. Lee, Y. Altunbasak, R. Mersereau, Multiframe error concealment for MPEG-coded video delivery over error-prone networks, *IEEE Trans. Image Process.* 11 (11) (2002) 1314–1331.
- [65] Y.O. Park, C.-S. Kim, S.-U. Lee, Multi-hypothesis error concealment algorithm for H.26L video, in: International Conference on Image Processing (ICIP), 2003.
- [66] Z. Yu, Z. Wang, Z. Hu, H. Li, Q. Ling, Video error concealment via total variation regularized matrix completion, in: International Conference on Image Processing (ICIP), October, 2012.
- [67] G. Gennari, G.A. Mian, L. Celetto, A H.264 decoder robust to transmission errors, in: Proceedings of EUSIPCO, vol. 1, 2004, pp. 114–120.
- [68] L. Su, Y. Zhang, W. Gao, Q. Huang, Y. Lu, Improved error concealment algorithms based on H.264/AVC non-normative decoder, in: Proceedings of International Conference on Multimedia and Expo (ICME), 2004.
- [69] S. Belfiore, M. Grangetto, E. Magli, G. Olmo, Spatiotemporal error concealment with optimized mode selection and application to H.264, *Signal Process. Image Commun.* 18 (2003) 907–923.
- [70] M.-H. Jo, W.-J. Song, Error concealment for MPEG-2 video decoders with enhanced coding mode estimation, *IEEE Trans. Consum. Electron.* 46 (4) (2000) 962–969.

- [71] S. Cen, P.C. Cosman, Decision trees for error concealment in video decoding, *IEEE Trans. Multimedia* 5 (1) (2003).
- [72] J. Koloda, J. Østergaard, S. Jensen, V. Sanchez, A. Peinado, Sequential error concealment for video/images by sparse linear prediction, *IEEE Transactions on Multimedia* 15 (4) (2013) 957–969.
- [73] X. Chen, Y. Chung, B. Changseok, Dynamic multi-mode switching error concealment algorithm for H.264/AVC video applications, *IEEE Trans. Consum. Electron.* 54 (1) (2008) 154–162.

# Introduction to Multimedia Signal Processing

# 10

Min Wu

*Department of Electrical and Computer Engineering and Institute for Advanced Computer Studies, University of Maryland, College Park, USA*

Multimedia information has become ubiquitous in our daily life, and its volume is increasing tremendously in the recent decades. This growth is a result of not just one technology area, but reflects the synergy and transformative advances of multiple technology fields. On the hardware side, multimedia devices have benefited from the steady advancement of VLSI semiconductor technologies, leading to increasingly more memories at a lower cost and enabling electronic systems to be built smaller and cheaper and run faster. Other contributing factors from hardware advances include newer and better displays, novel human-computer interfaces such as multi-touch technologies, and batteries of higher capacity and more compact in size. In addition, the worldwide development of high-speed network and wireless infrastructure has offered broadband channels—wired and wireless—enabling the exchange of information nearly anywhere at any time. With these technologies setting up the stage, multimedia signal processing provides core algorithms for efficient and effective representation, communications, analysis, understanding, and synthesis of multimedia information. Today’s smartphones and tablets have become a handy multimedia platform for all ages.

This section of the E-Reference presents three chapters offering an in-depth overview on several important aspects of multimedia signal processing, namely, the encoding and communications, the content-based search and retrieval, and the joint processing of audio-visual signals.

Multimedia data is becoming the dominant traffic in today’s wire line and wireless network systems. As with generic data transfer, a traditional paradigm to play video and/or audio located on a remote server is to transfer the entire media file to the receiver end and then playback the whole file. Although this download-then-play mode is possible for short media clips, the high data rate of multimedia makes it undesirable for long media content due to the long delays and large storage required. Streaming is a shift in paradigm to allow multimedia to be continuously received and presented to the end users; a media player can begin playing the data before the entire file has been transmitted. In the chapter on “Multimedia Streaming” [1], Greco et al. have provided in-depth discussions on how to stream multimedia data efficiently and reliably over various types of communication channels and networks. The authors analyze technical challenges arising from streaming over wired and wireless networks. Techniques addressing these challenges are presented, along with transport protocols from international standardization and emerging architecture.

With the proliferation of multimedia content came the need to efficiently organize and search multimedia content. In the chapter on “Multimedia Content-based Visual Retrieval” [2], Zhou, Li, and Tian

present a general framework on visual search that enables scalable retrieval of relevant content from large-scale visual databases. After an overview of the general search pipeline, the authors discuss key modules of the pipeline in details, namely, feature extraction, feature quantization, visual data indexing, retrieval scoring for index lookup, and post-processing. With the challenge of semantic gap in mind (originating from the difficulty connecting low-level visual features with high-level semantic concept), the authors introduce a variety of algorithms addressing the major issues in these modules.

Although each modality in multimedia data may be addressed on its own, the presence of multiple modalities together offers both challenges and opportunities for joint considerations. Thus a key general issue with multimedia signal processing is how to leverage beneficial aspects from several modalities and use them synergistically. In the chapter on “Joint Audio-Visual Processing for Video Copy Detection” [3], Liu et al. discuss joint audio-visual processing in the context of video copy detection. Video copy detection may be considered as a special class of search problems that focuses on very quickly finding near-duplicate copies related to a query video (instead of a semantically related one as for the more general search discussed in the previous chapter), and has applications ranging from content management to piracy deterrent. The authors review both audio and visual alone methods and joint audio-visual approaches. Based on these discussions, they present a video copy detection system to effectively fuse the detection results from both modalities, achieving more robust and accurate results.

I would like to thank all the authors for their contributions that have made this section possible. Several other experts in the multimedia areas also expressed interests in contributing to future editions. Multimedia is a broad area with multiple aspects interfacing other research and technology fields. To complement the current section, I would like to refer interested readers to the IEEE Signal Processing Magazine and related journals, where they can find quite a few articles offering informative overviews and surveys on such additional issues as multimedia security and forensics [4,5], multimedia social network [6], and more.

---

## References

- [1] C. Greco, I.D. Nemoianu, M. Cagnazzo, J. Le Feuvre, F. Dufaux, B. Pesquet-Popescu, *Multimedia Streaming*, Elsevier E-Reference on Signal Processing, vol. 5, Section 2, 2014.
- [2] W. Zhou, H. Li, Q. Tian, *Multimedia Content-based Visual Retrieval*, Elsevier E-Reference on Signal Processing, vol. 5, Section 2, 2014.
- [3] Z. Liu, E. Zavesky, D. Gibbon, B. Shaharay, *Joint Audio-Visual Processing for Video Copy Detection*, Elsevier E-Reference on Signal Processing, vol. 5, Section 2, 2014.
- [4] [E. Delp, N. Memon, M. Wu \(Eds.\), Digital forensics, IEEE Signal Process. Mag. 26 \(2\) \(2009\) \(special issue\)](#).
- [5] M. Stamm, M. Wu, K.J.R. Liu, Information forensics: an overview of the first decade, invited paper for the inaugural issue, *IEEE Access*, vol. 1, 2013.
- [6] [H. Bourlard, V. Krishnamurthy, Y. Sun, H.V. Zhao, K.J.R. Liu \(Eds.\), Signal and information processing for social learning and networking, IEEE Signal Process. Mag. 29 \(2\) \(2012\) \(special issue\)](#).

# Multimedia Streaming

# 11

**C. Greco, I.D. Nemoianu, M. Cagnazzo, J. Le Feuvre, F. Dufaux, and B. Pesquet-Popescu**

*Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI, 46 rue Barrault, 75634 Paris Cedex 13, France*

---

## Overview

In this chapter, we shall discuss the topic of multimedia streaming, a topic of uttermost importance in today's networking, dominated as it is by pervasive multimedia applications.

First, in [Section 5.11.1](#), we shall give a brief introduction to the multimedia streaming paradigm in general and with particular attention to video content. For the latter, we shall also introduce the basic concept of video coding, defining the general goals and constraints that a video coding system has to set up in order to provide a representation of the content suitable for streaming.

In [Section 5.11.2](#), we study the transport protocols ratified by international standardization bodies such as MPEG or IETF for video streaming. In particular, we focus on the MPEG-2 Transport Stream Protocol and the Real-Time Protocol, which introduced the techniques most commonly used in all media transport protocols.

In [Section 5.11.3](#), we analyze in more detail the challenges arising from a streaming transmission in a large-scale wired network, such as the Internet. We shall review the most relevant paradigms used nowadays, *e.g.*, the Content Distribution Network (CDN) approach, the Application Layer Multicast (ALM) approach, and the celebrated Peer-to-Peer (P2P) approach.

In [Section 5.11.4](#) we shall see how the streaming paradigm can be implemented in a highly dynamic and scarcely reliable environment such as a wireless network, with an overview of the most commonly used protocols, both in terms of general-purpose (*i.e.*, content agnostic) routing algorithms, and in terms of techniques tailored for the specific purpose of delivering multimedia content in real time.

In [Section 5.11.5](#), we shall present a framework of tools that can be employed in order to provide the multimedia stream with robustness toward the errors in its representation that may be generated during its transmission and to conceal their effects to the end-user.

In [Section 5.11.6](#), we discuss the encoding of high-quality video stream that also contain one or more subset streams at reduced quality, a technique to provide scalability to the video stream, so to make it adaptive with respect to the available bandwidth.

In [Section 5.11.7](#), we present the Multiple Description Coding framework, suitable when the streaming architecture presents, between the source and the destination of the stream, several available connections, all burdened with a non-negligible loss rate.

In Section 5.11.8, we present the multimedia streaming problem from a more innovative point of view. Namely, we analyze how a streaming architecture can evolve if the traditional routing protocol is set aside in favor of the more novel and flexible approach of network coding.

Finally, in the Conclusions, we discuss some open challenges and perspective in the field of multimedia streaming.

## 5.11.1 Introduction

### 5.11.1.1 Video streaming

During the last few decades, high-capacity broadband connections and high computational power resources have become more and more available to the average user for a relatively low cost. Jointly with the advancements in the field of digital signal compression, this has caused a paradigm shift of the majority of services provided over the Internet, from text-based to multimedia.

Looking at the evolution of the Internet, we observe that the first *killer applications*, *i.e.*, those applications that have been in themselves a sufficient reason to use the Internet, have been text-based. The wide adoption of the Internet as a global communication and information-exchange tool took place through the introduction of the e-mail and chat services, that quickly interconnected people for both business and leisure; the bulletin board service, that first allowed users to upload and download software and data, post and read news and bulletins; the web browsing, for news and information retrieval, online shopping, and—more recently—social networking.

With the subsequent larger diffusion of broadband connections, jointly with the introduction of the Peer-to-Peer paradigm, the users have grown progressively accustomed to multimedia services—at the beginning, mostly music (in the celebrated MP3 format [1,2]) and images (in JPEG and JPEG 2000 [3,4]). All this belongs now to the past: nowadays the Internet is dominated by video content.

Apple Inc., one of the largest publicly traded company in the world by market capitalization and one of the first technology company in the world by profit, has reported to have sold over 30 million of videos through its iTunes Store just between 2005 and 2006.<sup>1,2,3</sup> Nowadays, it has sold over 200 million TV episodes, over 1 million of which in high definition, and several million of feature-length movies.<sup>4</sup>

During the same period of time, the popular video hosting service offered by YouTube has passed from serving about 100 million video viewings *per day* in 2006 to 1 billions in 2009, 2 billions in 2010, and over 4 billions today.<sup>5,6</sup>

Moreover, according to the *Global Internet Phenomena Report*, Netflix Inc., the most successful American provider of on-demand Internet streaming media with 24 million subscribers in the United States and over 26 million worldwide, is nowadays the largest source of North American Internet traffic, accounting for almost 25% of the aggregated traffic.<sup>7</sup>

<sup>1</sup>Neil Hugues, *Apple now worth more than Google and Microsoft combined*, AppleInsider.com, February 2012.

<sup>2</sup>Ben Rooney, *Apple tops Exxon as most valuable company*, CNN, January 2012.

<sup>3</sup>Jim Feeley, *Video everywhere*, PC World, April 2006.

<sup>4</sup>Apple Press Info, <http://www.apple.com/pr>, Consulted in April 2012.

<sup>5</sup>Reuters, USA Today, *YouTube serves up 100 millions videos a day on-line*, July 2006.

<sup>6</sup>Website Monitoring, <http://www.website-monitoring.com>, Consulted in April 2012.

<sup>7</sup>Sandvine Inc., *The Global Internet Phenomena Report* [http://www.sandvine.com/news/global\\_broadband\\_trends.asp](http://www.sandvine.com/news/global_broadband_trends.asp), Autumn 2011.

The volume of traffic over the Internet due to video content has been growing exponentially, and is expected to continue growing. David Hsieh, vice-president of worldwide leader designer, manufacturer, and sellers of networking equipment Cisco Systems Inc., for Tele-Presence and Emerging Technologies, has pointed out in an interview that “video is invading all aspects of our lives [...]. Today, over half of all Internet traffic, 51%, is video.” According to his estimations, in less than three years, digital video content will account for 90% of the aggregate Internet traffic.<sup>8</sup>

These days, video options are proliferating at an astonishing pace: everything, from Hollywood movies and TV shows to clips from ordinary users, is available to whomever is connected to the Internet, whether with a laptop, a notebook, or a tablet. More recently, even mobile phones—built with advanced computing ability and connectivity and commonly referred to as *smartphones*—can constantly access video content with news, sports, and video segments. The new frontier of network communications lies in this new paradigm: “Video Anywhere at Anytime”.

Even though these applications are becoming nowadays commonplace, and the technology involved has greatly advanced during the past few years, a great deal of further improvement is still needed before Internet streaming can consolidate its status as viable alternative to traditional television broadcasting modes. These improvements involve many technical challenges both in the domain of video coding and of networking [5].

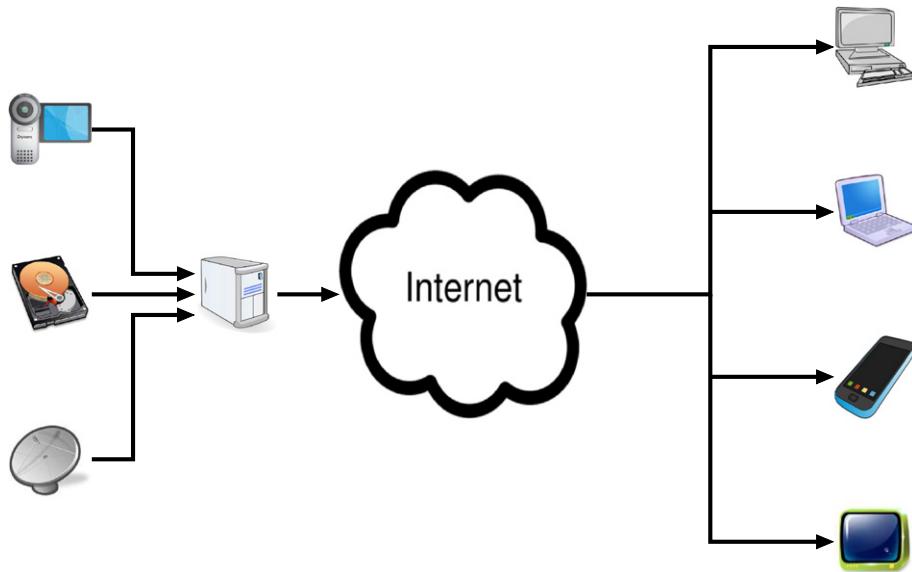
To give a formal definition, video streaming is a video content transmission paradigm in which the content is continuously received and presented to the end-user, while being delivered by the content provider. It is from this continuous playback, which distinguishes streaming from download-and-play schemes, that most of the design challenges arise. Figure 11.1 shows a simple example of video streaming architecture.

Streaming differs in general from video-telephony and videoconferencing services, which also transmit in real-time, as in video streaming the content might be encoded without the benefit of knowing the state of the channel during transmission [6]. Furthermore, streaming is distinguished by its ability to store media data encoded off-line, and by its tolerance to a longer playback delay. If the distribution type is such that users request a video content that is already encoded and stored on a server, the service is referred to as *on-demand streaming*. If by its nature the content cannot be pre-encoded and stored on a server, such as a sport or news event to be broadcast live, the service is usually referred to as *live streaming*.

In a media streaming system, the media server packetizes the content into data units in order to transmit them, on demand or following a time schedule, for playback in real time. The data units are delivered from the streaming server to the clients using a transport protocol, such as RTP, while the clients may interact with the server using a control protocol, such as RTSP. The clients bufferize the data they receive and begin the playback after a short delay. This delay is introduced to let the client collect an amount of media sufficient to prevent the effect of packet losses and delays from constantly interrupting the play-out of the stream. The choice of the right buffering delay involves a trade-off between the reliability of uninterrupted play-out, the memory resources of the player application, and end-to-end delay itself. It is usually fixed and not depending on the length of the presentation. In most commercial media streaming services (such as Real Player, available at <http://www.realnetworks.com>, and Windows Media Player, available at <http://www.windows-media.com>), which typically operate at

---

<sup>8</sup>Andy Plesser, *Cisco: Video will Account for 90% of Net Traffic in Three Years*, October 2011.

**FIGURE 11.1**

A simple example of video streaming architecture. The video content, may it be available as a live feed, a stored file, or a satellite communication, is processed and stored on a server, which transmits it through the Internet to a multitude of heterogeneous clients.

50–1500 kbps, the optimal balance is found in the range 2–15 s [7]. All the transport-related aspects of a streaming service will be discussed in detail in [Section 5.11.2](#).

If some data packets are lost, according to the specific application, the data unit may or may not be sent again in another packet [6,8]. As we shall see in detail in [Section 5.11.5](#), if the application chooses not to send the packet again, then the end-user may try to reduce the impact of the loss using an *error concealment* strategy, which can be based on redundancy added at the server for this purpose (*forward error concealment*), on characteristics of video signals itself (*error concealment by post-processing*), or on a dialog between the server and end-user (*interactive error concealment*) [9].

The inherent requirement of continuous delivery translates to a continuous connectivity requirement between the media server and clients. In many applications, it would be also desirable to have a graceful degradation of received media quality as network environment resources change over time [10,11].

### 5.11.1.2 Video coding

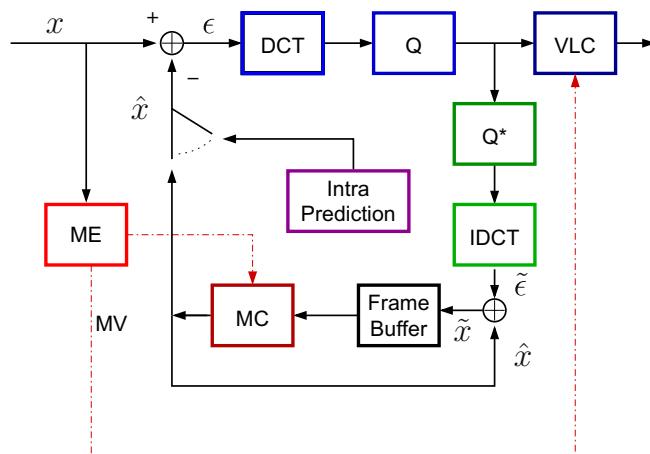
Video compression or coding is an almost mandatory step in storage and transmission of video, since uncompressed video has overwhelming resource requirements. However, video is a highly redundant signal, characterized by the similarity among different frames (also known as temporal redundancy), and homogeneity within any single frame (spatial redundancy). In virtually all compression systems both kinds of redundancy are exploited in a spatial compression stage and a temporal compression stage.

The most successful video compression schemes to date are those based on hybrid video coding. This term refers to the two different techniques used in order to exploit spatial redundancy and temporal redundancy. Temporal compression is achieved by computing a motion-compensated prediction of the current frame and then encoding the corresponding prediction error. Of course, such an encoding scheme needs a motion estimation stage in order to find motion information necessary for prediction. Spatial compression is obtained by means of a transform-based approach, which makes use of the discrete cosine transform (DCT), or its variations. However, since the introduction of the optional modes in H.263+ [12] and of the Intra modes in H.264/AVC [7, 13] and MPEG-4 p.2 [14], the spatial compression benefits from both (spatial) prediction and transform.

We will look into the video compression process at two levels: the block level and the image level. We remark that these structures have different names in the standards: coding unit, macroblock, block, subblock for a matrix of pixels; coded picture, frame, for an image. However we will keep the neutral terminology of block and image in the following.

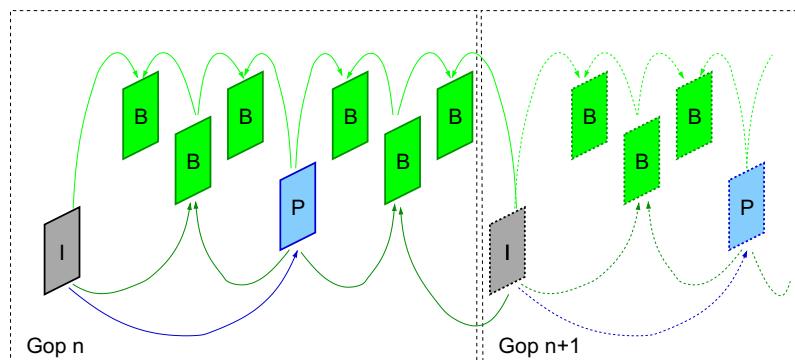
At the block level, we consider the general scheme of a modern hybrid encoder given in Figure 11.2. The current block of pixels  $x$  can be coded using two families of modes: Intra and Inter. In an Intra mode, the block is encoded without any reference to other frames, so it can be decoded independently from them. However, the current block can be predicted from previously encoded blocks of the same frame: in this case the prediction  $\hat{x}$  is produced by the Intra Prediction stage. Commonly, neighboring pixels are used to create the prediction of the current block: they can be averaged to produce a constant-valued prediction (DC prediction) or copied along some given direction (Directional prediction). Once the prediction obtained, the residual  $\epsilon = x - \hat{x}$  is encoded with a JPEG-like algorithm, consisting in a spatial decorrelating transform (typically DCT or its variants), a quantization ( $Q$  in Figure 11.2) and a variable length coding (VLC).

In the Inter mode, the current block  $x$  is predicted by motion compensation from previously encoded frames: the prediction  $\hat{x}$  is a block of pixels having the same size and shape of  $x$ , but belonging to another



**FIGURE 11.2**

General scheme of a modern hybrid video encoder.

**FIGURE 11.3**

A GOP structure with  $N = 8$  and  $M = 3$ .

(already encoded and decoded) frame. In order to the decoder being able to recreate this prediction, the encoder should send a motion vector pointing to the position of  $\hat{x}$ , and an index identifying the reference frame within a reference list. This index was not needed in previous standards such as MPEG-2, where there was only one possible reference frame. The task of finding the best temporal predictor among the possible candidates is called motion estimation. A variation of the Inter mode is the bi-directional mode, where two reference lists (or two images) can be used to create the prediction of the current block. Just like for the Intra modes, after creating the prediction  $\hat{x}$ , the residual is computed and encoded by spatial transform, quantization, and entropy coding. Finally we observe that the encoder stores locally the decoded frames, obtained by adding the prediction to the reconstructed error, produced by dequantization ( $Q^*$ ) and inverse spatial transform (IDCT). The decoded frames are used in order to produce the temporal prediction. We observe that this prevents the drift effect between encoder and decoder.

We consider now the compression process at the frame level. In current video standards, there are mainly three kind of frames, referred to by the letters I (for intra), P (for predictive), and B (for bi-directional). I and P images are sometimes called anchor frames, and we will follow this convention in the remainder of this section. The images of the video sequence are organized in a periodical structure called group of pictures (GOP), in which the first image is always an I frame. The GOP structure is defined by the number of images  $N$  and the number of B frames  $M$  between two anchor frames. In Figure 11.3 we show a GOP with  $N = 8$  and  $M = 3$ .

Intra frames have a strong constraint: all the blocks in them should be encoded in intra mode. Therefore they typically require a much higher rate than a P or a B frame for achieving the same quality. Nevertheless they provide some very important functionalities to the encoded video, namely:

**Random access.** An intra frame can be decoded independently from other frames, so it can be used to access to a random part of an encoded video sequence.

**Error propagation limitation.** If the encoded image received by the decoder is affected by transmission errors, all images which are predicted from it are potentially affected by errors. Periodically stopping the prediction process puts an higher bound on the error propagation.

**Reduced frame-rate decoding** of fast-forward playing. It is possible to decode and play only the I frames of an encoded video, thus allowing a faster video reproduction without needing a

faster decoder, since images are simply skipped. This is a form of temporal scalability (see also [Section 5.11.6](#)).

Predictive frames allow Inter and Intra blocks, but not the bi-directional ones. They have typically better compression performance than I frames, but they are more computationally demanding, because of the motion estimation process. Bidirectional frames are the most effective in terms of rate-distortion performance, but also the most computationally expensive, since they require twice the motion estimation process than P frames.

In conclusion, independently from the actual video compression standard, the GOP structure highly impacts the characteristics of the compressed video with respect to the streaming applications: using less Intra frames allows a rate reduction, but in the case of errors, more time will be needed before being able to decode the video again; P frames provide a good trade-off between complexity, compression, and error propagation; however the best compression performance can only be reached if also B frames are used, introducing however encoding and decoding delays, and requiring larger buffers. Therefore the choice of the GOP structure has to be done very carefully. At this end, modern standards such as H.264/AVC allows several improvements, by the means of new image types (SI and SP, described in [Section 5.11.5](#)) and of a more flexible dependency structure. Moreover, the need of tools for progressive (or scalable) video coding has pushed for the creation of a scalable video coding amendment, which will be described in [Section 5.11.6](#).

## 5.11.2 Transport protocols

### 5.11.2.1 Overview

During the development of video streaming services over the Internet in the past two decades, a large number of transport protocols have been experimented by the major actors of the area. In this section, we will restrict our study to protocols ratified by international standardization bodies such as, MPEG<sup>9</sup>, or IETF<sup>10</sup>, however all media transport protocols use common techniques to deal with the constraints implied when deploying a video streaming service, as seen in [Section 5.11.1](#). We will especially study two protocols:

**MPEG-2 Transport Stream** (or MPEG-2 TS) [15] is the standard used throughout the world for Digital TV (DTV) systems. It is used for cable, satellite, digital terrestrial TV, and IPTV. The protocol has become a cornerstone of the DTV industry, and most of its infrastructure uses MPEG-2 TS based equipments. The protocol is designed for pure broadcast and does not rely on any return channel.

**Real-Time Protocol** (or RTP) [16] is the standard used world-wide for real-time communication over IP networks. It is used in audio or video conferencing systems, some video-on-demand or live TV systems and can be deployed in point-to-point services as well as point-to-many and many-to-many services, usually using IP multicast. Although mainly intended for connected devices, the protocol may also be used in broadcast environments.

<sup>9</sup> <http://mpeg.chiariglione.org/>.

<sup>10</sup> <http://www.ietf.org/>.

Apart from these protocols designed for real-time media transport, a growing number of media delivery systems, especially video on demand systems, now rely on the TCP layer to deliver media files over the Internet. A very large number of media file formats exists, and most if not all of them can be carefully authored so that media playback can start while the file is being downloaded; if bandwidth is sufficient and no seeking in the media timeline occurs, the playback can happen without any stalling. Because this provides an experience close to what is achieved by stream-based protocols (RTP, MPEG-2 TS), such file delivery systems are also labeled as streaming protocols.

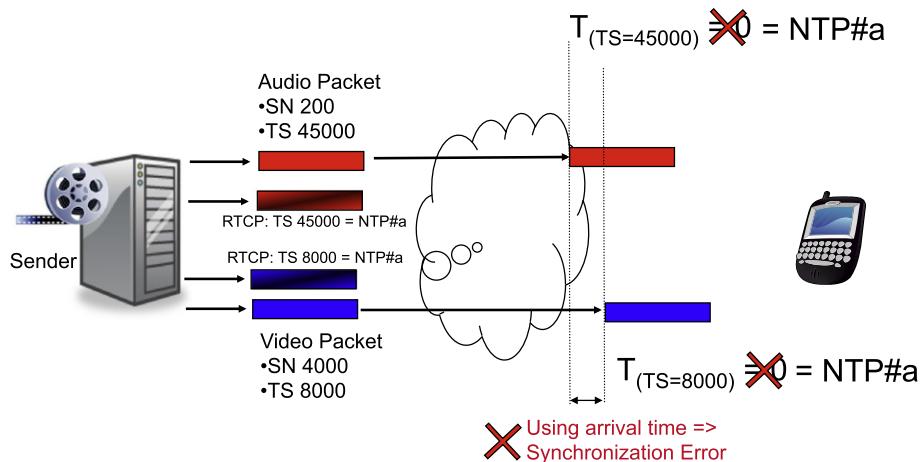
### 5.11.2.2 Synchronization

The first problem to solve in media transport is to ensure a proper playback of a media stream, such as an encoded video. Video frames are typically captured at a constant frame rate (for example 24 frames per second for movies), but it may happen that some frames are dropped during the capture or encoding process, as is often the case when using Web cameras. On the other hand, some media such as subtitles are not captured at a fixed rate and usually their frames have very different durations. For these reasons, relying on an index to indicate when the media frame shall be presented to the end-user is not sufficient.

In order to provide generic mechanisms ensuring accurate timing of the media frame during playback regardless of the media type, all streaming protocols introduce the notion of timestamp, a timing value associated with each media frame of the stream. This time information may be expressed in the sampling frequency of the stream, for example by choosing a multiple of the sample rate for audio or of the frame rate for video; some protocols may enforce a given frequency, such as 90 kHz for video timing. The terminology may vary between protocols, such as TimeStamp, Presentation TimeStamp, or Composition TimeStamp, but the meaning is the same: the value represent the instant in time where the media data is audible/visible by the viewer. This mechanism is referred to as *intra-stream synchronization*: once the first media frame  $F_0$  with timing  $T_0$  is presented to the user, all the other frames  $F_k$  can be presented at the right time by ensuring that the delay  $T_k - T_0$  is respected, even if some frames between  $F_0$  and  $F_k$  were lost during the network transmission.

The second step in enabling a proper playback is to ensure that multiple media streams are presented together in a synchronized way, for example in lip-sync audio or closed captioning services. This *inter-stream synchronization* is achieved by defining a common time base or clock that all timestamps in the media stream refer to. Transport protocols usually handle this in their own way; in MPEG-2 TS, all timestamps are expressed in 90 kHz, start from the same origin in time, and therefore can be directly compared to each other. Additionally, the time itself is sent within the bitstream through PCR (*Program Clock Reference*) timestamps, expressed in 27 MHz; sending the clock avoids any time drift between the decoder and the encoder notion of time, MPEG-2 decoders being non-connected devices.

RTP uses a slightly different approach: since the standard was conceived for audio-visual conferencing, the streams received at the client side may not come from the same network source, and consequently synchronizing the timestamp origin of each RTP stream would be quite difficult; therefore, each media stream uses its own frequency and timestamp origin, and a special packet is sent on regular basis to associate the RTP time of the stream to an NTP (Network Time Protocol [17]) value. As illustrated in [Figure 11.4](#), by comparing the NTP of each stream, the client can synchronize them precisely, at least with the same precision as the NTP precision of the originating sources. Note that it is not possible to synchronize RTP streams using only RTP timestamps, as packets do not share the same timeline origin and may suffer different delays during the transport.

**FIGURE 11.4**

RTP inter-stream synchronization. By explicitly associating an NTP value with the RTP timeline, synchronization between streams can be achieved and delays in the transmission network can be estimated.

### 5.11.2.3 Transmission delay

As seen in [Section 5.11.1](#), IP networks induce transmission delays, and most problematically jittering, *i.e.*, variations in the transmission delay. For example, a client buffering two seconds of media to ensure proper synchronization will likely experience a buffer underflow (no media to present) when the jitter gets above two seconds; in this case, the playback will have to be paused until enough media data is available. Obviously, the larger the buffer is, the less likely a stalling in playback will occur, but this implies a longer initial waiting time (filling up the buffer) and a higher memory consumption. This buffer is usually referred to as the de-jittering buffer. The size of this buffer depends on the network characteristics and on the application requirements; low-latency bi-directional communication, such as video or audio conferencing, will require the smallest de-jittering buffer possible in order to avoid high delays in the reception and decoding chain, which degrade the service quality; a decent user experience for conversational services is usually achieved with an overall round-trip delay of 300–800 ms, *i.e.*, 150–400 ms to perform media capture, encoding, transmission, de-jittering, optionally packet retransmission, decoding, and presentation to the user. It is usually considered that 150 ms is the maximum tolerable downlink delay for hard synchronous services, such as real-time medical imagery services [18].

Unidirectional communications, such as on-demand or live streaming services, may on the other hand use a larger buffer to increase the service quality, avoiding stalling of the playback and using the delay provided by this buffer to recover lost packets if any. The presence of the de-jittering buffer implies that the amount of data present in the client but not yet decoded, usually referred to as the decoding buffer, varies a lot, depending on the network conditions, application requirements, and terminal implementation; consequently, there could be times where the buffer contains more data than initially planned, for example when the downlink delay decreases. It is therefore important that incoming data can be stored, regardless of the decoding/de-jitter buffer occupancy. Streaming models where decoding buffers cannot overflow are called *loose buffer models*, and are quite common on IP networks.

MPEG-2 TS, on the other hand, does not consider variable delay in the transmission chain; although it can be used on variable bitrate links, MPEG-2 TS design is based on the principle that encoded media bits leave the multiplexer/enter the demultiplexer at a constant rate. Moreover, the MPEG-2 TS protocol, designed in 1992–1993, takes into account the constraints of existing consumer electronics hardware, including low-memory usage. The protocol therefore follows a strict buffer model, where the amount of memory available to store media data before decoding is fixed. This implies that when the decoding buffer is full, any incoming data will be lost, causing a buffer overflow. It is the responsibility of the encoder or multiplexing system to make sure that the decoder removes data from the decoding buffer at the right timing to avoid this buffer overflow; in order to do so, MPEG-2 TS precisely describes how bits are removed from the decoding buffer based on their decoding time, given by means of a decoding timestamp carried along with the presentation timestamp in the bitstream. The encoder can therefore manage how many bits are available in the decoding buffer at any time, and adjust its bit allocation policy without risking any overflow. This decoding timestamp is specific to strict buffer models, and is not used in RTP-based streaming protocols.

#### 5.11.2.4 Transmission errors

When deploying a streaming service, transmission errors need to be considered. These errors mainly depend on the underlying physical network, but common techniques are used to detect these errors and help the client recover. Note however that not all protocols or applicative deployments of these protocols require the same tools to handle transmission error recovery, or to monitor the quality of the service. In order to detect packet losses, the common approach is to embed in each packet of the media stream a counter incremented at each packet, thereby allowing a client to detect packet losses. Both RTP and MPEG-2 TS use this tool, with however a major difference: in the case of RTP, the counter has a much larger span (16 vs. 4 bits); because RTP is conceived to run over UDP/IP, it is clear that the order of delivery cannot be guaranteed: packets may arrive out of order due to some router being overloaded and the network path being changed afterward; the large span of the RTP counter is designed to reorder packets at the receiver side, and all compliant RTP receivers shall implement this RTP reordering mechanism. MPEG-2 TS on the other side was conceived for bit-serial transmission, which guarantees that packets are always received in the order they were sent.

When an uplink channel is available, error recovery usually starts with the monitoring of various parameters such as network jitter or packet loss rate; for example, the RTP protocol includes in its specification a companion protocol, RTCP or *Real-Time Control Protocol* [16], in charge of exchanging various quality parameters between the receivers and the sources through *sender and receiver reports*: each source may indicate how many packets were sent since last report, what is the correspondence between RTP time of the media and the NTP or when the report was sent; each receiver emits reports indicating how many packets were lost, when the last report was received, when this report was sent, and may include more advanced information such as whether burst errors were detected or which packets were lost or duplicated; it may include application-specific metrics such as processing delay between the network card buffer and the presentation of the media. Such a monitoring may then be used by the sender to try to improve the service quality, whenever possible, with adjustments in Forward Error Correction (FEC), in media bitrate or in delivery delay for on-demand services. It is however important to limit the traffic of the monitoring, (in the case of RTCP, maximum 5% of the overall RTP traffic bandwidth),

and not send a report for each media packet received; in RTCP, this is enforced by scheduling reports with a minimal period of 5 s.

When no uplink channel is available from the receiver to the source, or whenever monitoring of clients is not meaningful or cannot be performed in real-time (*e.g.*, one-to-many multicast of a live event), error recovery is usually addressed by generic error correction techniques or by using media-aware concealment tools, as will be seen in [Section 5.11.5](#). Generic correction techniques include packet retransmission and usage of FEC data at the application level, *i.e.*, the FEC data is treated like the media data and both media and FEC data are handled at a layer above the transport protocol. Although all these techniques can easily be deployed, they all have their pros and cons.

FEC may correct some of burst errors, but requires more bandwidth to be sent as well as more processing power; FEC is a solution that scales well with large number of recipients, since the FEC data is the same for everyone; on the other hand, the FEC itself may be corrupted due to packet loss and may require a large data block to be efficient, thereby increasing the end-to-end delay. A good example of FEC processing can be found in the DVB-H standard [19] for Digital Broadcast TV on mobile devices, where media data is protected by Reed-Solomon [20] codes; the protected transmission technique, known as MPE-FEC [21], involves gathering chunks of 2 Mbits of IP packets before applying the FEC. Another similar example can be found in the FLUTE protocol, used to deliver files over IP multicast networks. FEC is also used in IPTV core networks, where MPEG-2 TS packets are carried over RTP.

Packet Retransmission implies a back channel and must happen in a short time frame, *i.e.*, less than the de-jittering buffer depth, in order not to impact the service quality. The main drawback of this technique is its inability to scale with the number of users: since different receivers most likely loose different packets, the number of retransmission requests linearly increases with the number of receivers (assuming the same error rate), inducing a potentially very high traffic just for the retransmission requests; sending a request for each lost packet could result in overuse of the available monitoring bandwidth. RTCP has relaxed rules [22] allowing *emergency feedback* to be sent by the receiver in order to signal a packet loss without having to wait for the next scheduled report, while still maintaining the 5% bandwidth constraint. It is important to notice that the actual retransmission decision is left up to the sender, as only the source may know whether it is possible or useful to send the lost packet(s) back.

MPEG-2 TS, on the other hand, has been designed for unidirectional broadcast networks, with no uplink data available: the protocol provides all the tools required to identify packet losses (packet counters, polynomial checksum such as CRC), but does not provide mechanisms for packet retransmission. With the deployment of IPTV services, the usage of MPEG-2 TS over IP networks has become a key part of the infrastructure, and error recovery for MPEG-2 TS had to be addressed; the solution used by the industry [23] is to carry an integral number of MPEG-2 TS over one RTP packet, thereby reusing UDP reordering capabilities of the RTP header as well as the possibility to integrate with well-defined FEC tools; in such deployments, the RTP timestamp represents the target transmission time of the first byte of the first TS packet in the payload, but is most of the time ignored by receivers.

### 5.11.2.5 Bandwidth adaptation

One of the main issues faced when deploying real-time video services over IP networks is the available bandwidth variation. A sudden decrease in network capacity will result in a high packet loss rate if the media bitrate is not adapted to the new available bandwidth. This challenge has been identified as early

as the first experiments of real-time video streaming were conducted, and various solutions have been proposed to adapt the bandwidth of the media to the network conditions. Since encoding on the fly the media data to provide hundreds of bitrates for thousands of users requires tremendous computing power hence deployment costs, a common approach used in the industry is to provide the media to be delivered at different bitrates, either by reducing the video resolution, the frame rate or the quality, or a combination of them. Powerful, state-of-the-art solutions to this problem are addressed in Sections 5.11.6 and 5.11.7. The server (resp. the client) is then responsible for providing (resp. requesting) the appropriate media bitstream according to the available bandwidth. Such a technique is usually referred to as stream switching [24], and has several drawbacks, such as requiring high storage capacity and adding complexity of switching between bitrates in a seamless fashion, *i.e.*, unnoticeable for the viewer. Since the receiver does not know the specificities of the coded media, in order to achieve seamless switching, it shall either:

- ask the server to switch whenever possible, *i.e.*, at GOP boundaries or when media-codec specific switching points are present,
- stream two versions of the content at the same time and figure out by itself when switching is possible.

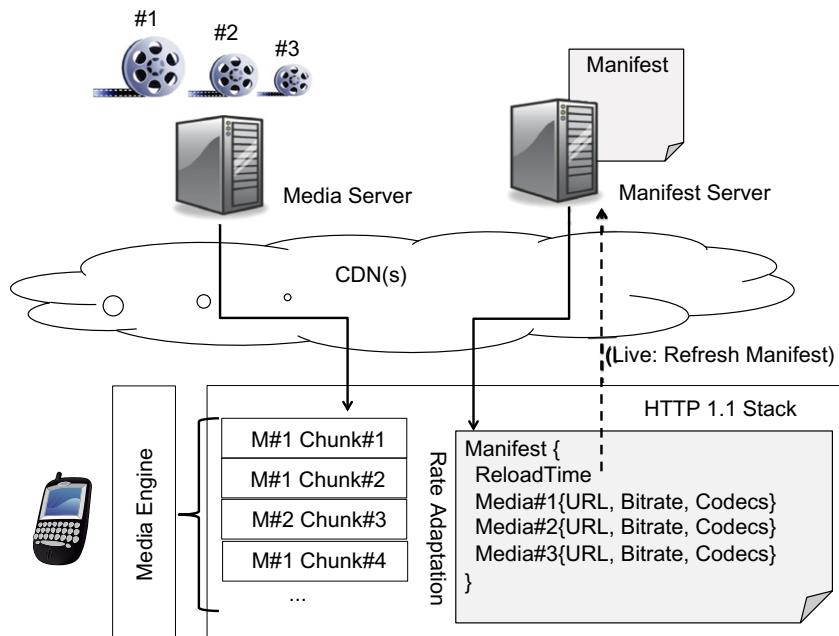
The second solution is not an option, as streaming two bitstreams in parallel would require much more bandwidth than usually available at the client side. The first option is usually what is implemented in commercial deployments, however most of the time the server is capable of deciding itself, based on RTCP reports, whether the bandwidth should be switched or not.

Until the mid-2000, deployments of video services over IP for the end-user were heavily relying on ad-hoc network infrastructures: because of real-time constraints, most solutions were using UDP, which did not always guarantee to be available between the source and the receiver (although the situation has now changed and UDP traffic is more and more available). Moreover, the solutions were using ad-hoc servers, mostly based on RTP/RTCP streaming, which require dedicated physical machines to deploy as close as possible to the edge of the network. The advent of large Content Delivery Networks (CDN) and the success of video downloading services such as YouTube have marked a turning point in streaming protocols, moving from a server-driven logic (RTP and RTCP) to a client-driven logic, with a large family of HTTP Streaming protocols [25] mostly driven by Microsoft Smooth Streaming [26], Apple HLS [27], or the MPEG-DASH standard [28].

The principles of these protocols are illustrated in Figure 11.5 and can be summarized as follows. The key point is to make media data accessible per large chunks rather than per packet as in RTP or MPEG-2 TS; chunks can be fetched as files through HTTP if they are in independent files on the server, or with HTTP 1.1 byte-range requests if they are part of a larger file on the server. The advantage of this technique is that all media data is exchanged through standard HTTP requests, and this allows deploying a video service over the existing Internet network, or Over-The-Top (OTT), including all the content caching policies deployed within CDNs. By turning video streaming services into a simple file download protocol, deployment costs are drastically reduced. Chunks are usually based on well-supported media formats, such as ISO Base Media File [29], WebM,<sup>11</sup> or MPEG-2 TS.

---

<sup>11</sup> <http://www.webmproject.org/>.

**FIGURE 11.5**

Design of HTTP streaming protocols. The client is in charge of asking each chunk to the server, performing the rate adaptation and refreshing the list of chunks to play in live mode. The distribution network (servers and cache) is only HTTP based.

In order for the client to download only what it needs for the current time slot, these protocols describe as much as possible the chunks and properties of the media data in a concise way (indexing process) and give this information to the client through an HTTP server, again as a file exchange; this file is usually referred to as the *manifest file*. With this information, the client can then download only the chunks it is interested in (for example when seeking), or even only the first key frame of each chunk when doing trick modes such as fast forward or rewind. In order to adapt to varying network conditions, these protocols reuse the stream switching mechanism seen previously, and provide as many encodings as deemed necessary; the client then selects the most appropriate chunk for its current available bandwidth. The bitstream-switching process is simplified by enforcing that a client may perform switching from chunk  $N$  of encoding  $A$  to chunk  $N + 1$  of encoding  $B$ , for any  $A$  and  $B$  encoding; in practice, this is usually achieved by starting a new GOP at the beginning of each chunk, or sometimes by enforcing alignment of GOPs in all encodings. With all these tools in place, a client can perform bitrate adaptation of the media whenever necessary; adaptation logic may be driven by available network bandwidth, by CPU or memory usage, by viewing conditions as such thumbnail video versus full screen video or play speed, etc. Since these protocols are HTTP-based, i.e., file-based, they had to define special modes for delivering live content, which per definition is never ending and cannot be encoded in a single, finite

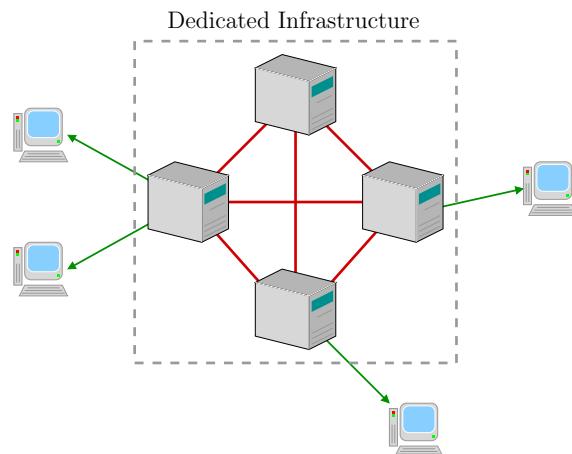
size file as needed by HTTP servers. This limitation is overcome by describing the live as an infinite succession of chunks of a given duration. For a time given, chunks describing the period from  $T$  to  $T + D$  seconds are listed in the manifest, along with a refresh date. When the current time gets close to the refresh date, the client downloads the manifest from the HTTP server and gets an updated list of chunks describing  $(T + D, T + 2D)$ . The HTTP Streaming encoder can then decide if the past chunks should be removed from the server, or archived for later on-demand viewing.

### 5.11.3 Streaming on wired networks

Multimedia streaming over large-scale wired networks has been widely investigated over the last few years. Traditionally, multimedia content has been delivered using the client/server paradigm, which is currently being abandoned, as a server alone might not be able to handle a large number of clients. Nevertheless, the client/server paradigm is still employed; *e.g.*, YouTube uses it for low popularity videos [30].

#### 5.11.3.1 Content Distribution Networks

Content Distribution Networks (CDNs) are also used for distributing multimedia content [31]. They inherit from the client/server model, but employ multiple servers interconnected through a dedicated infrastructure, partitioning the large load [32]. YouTube, for instance, employs the CDN of Akamai to deliver popular videos. Architecturally as illustrated in Figure 11.6, CDNs are large-scale infrastructure networks, composed of powerful servers that can handle multiple connections, geographically dispersed to reduce the total traffic on each server, and interconnected by high bandwidth dedicated links.



**FIGURE 11.6**

Architecture of a Content Distribution Network.

### 5.11.3.2 Application Level Multicast

Nowadays, a new paradigm for multimedia content distribution is emerging: video multicast. Two types of multicast are considered: IP multicast and Application Level Multicast (ALM).

IP multicast takes advantage of the implementation of multicast functionalities in the network layer, as at these levels the network topology is best known. Unfortunately, IP multicast suffers from scaling issues and a lack of support for higher layers functionalities. It also requires costly infrastructural changes, since the routers have to integrate IP multicast [33]. ALM is thus favored for its low cost and its simplicity of implementation, which does not require any infrastructural changes.

### 5.11.3.3 Peer-to-Peer distribution

An interesting alternative for multimedia content distribution is the Peer-to-Peer (P2P) paradigm, which provides inherent self-organization and resource scalability properties [34]. The P2P model is a data transmission paradigm wherein, unlike the client/server model, there is no central server. When a user wants to obtain a content, it contacts another user (referred to as peer) that already has the content available. The multiple benefits of this simple yet revolutionary idea include are load reduction, resource aggregation, and increased reliability. Several applications benefit from this paradigm, such as file sharing, personal communications (instant messaging, audio and video conferencing, *etc.*), user collaboration (cooperative document editing, on-line gaming, *etc.*), and distributed computing.

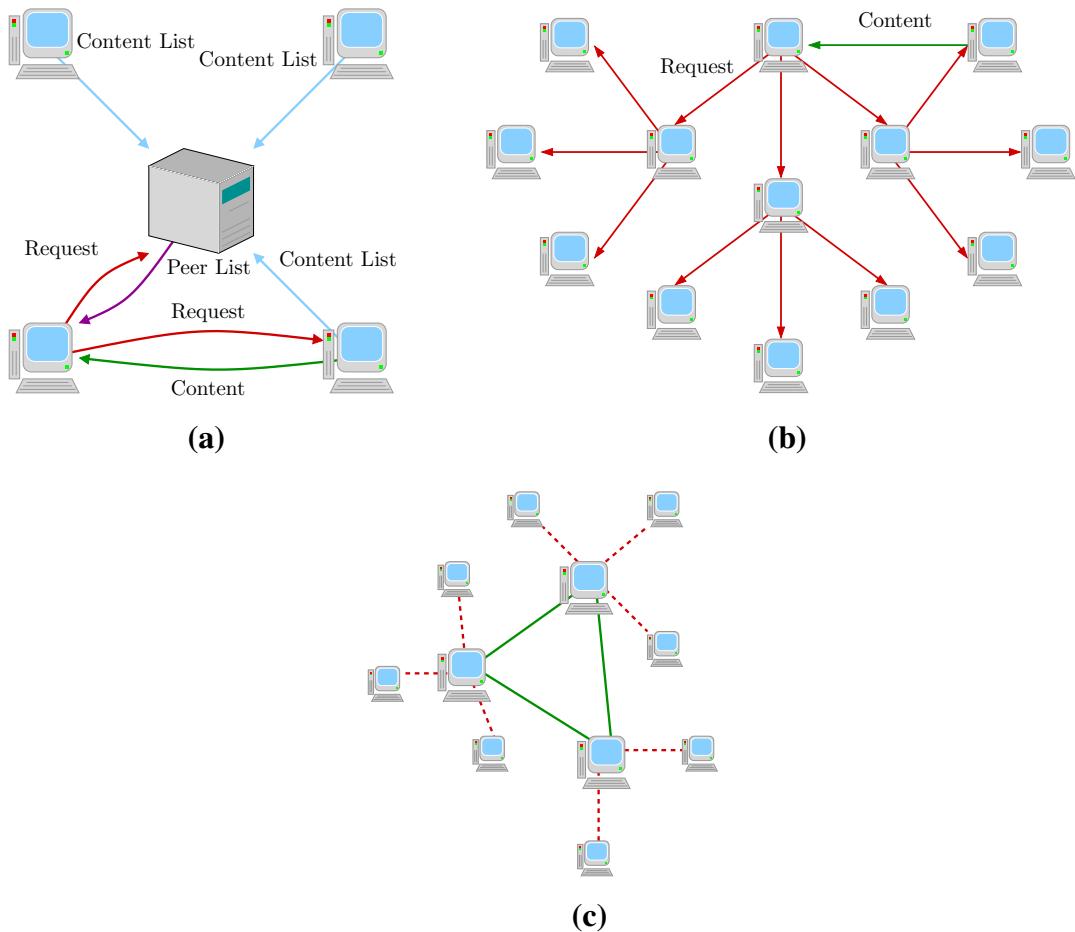
Even though its widespread adoption is relatively recent, Peer-to-Peer is not a new concept. In fact, in the 1960s, the Internet was already rather focused on data sharing among small groups of companies and universities (ARPANET). However, the transmission cost at the time and the volume of data was too high to fully exploit potential of P2P.

In 1999, with the introduction of Napster, P2P started to have a relevant impact on Internet traffic. Napster was the first large-scale P2P network for content sharing, mostly music in MP3 format, with more than 250,000 users. Napster was shut down in July 2001 due to copyright issues. Nevertheless, during the following years, several similar networks have arisen, the most relevant being Gnutella (2000), Kazaa (2001), eDonkey (2003) and, more recently, BitTorrent.

In a P2P network, the politics that determines the way a peer is connected to another peer having available the requested resource is referred to as architecture. P2P architectures can be broadly classified in three families: centralized, decentralized, and hybrid.

In a centralized architecture (Figure 11.7a), such as the one used by Napster, peers send their requests to a central server, where a list of all connected peers and the resources they control is stored. The server provides a list of peers having the requested content available. Once the list is received, the requesting peer chooses one of the peers on the list and contacts it to set up the transfer. After the transfer is completed, the peer communicates to the server that it now has the resource, so that the server can update its database. The main advantage of centralized architectures is the ease of locating the content: searches are simple and inexpensive. The principal drawback is the presence of a critical single point of failure: if the server suffers a malfunction, the entire network is affected. Moreover, anonymity is not preserved, as the identity of each peer and the content it provides is stored on the server.

To partially overcome the former problem, a variant of this architecture has been introduced by eDonkey, which makes use of multiple servers. In this architecture, different groups of peers connect to

**FIGURE 11.7**

Architectures of P2P networks. (a) Centralized. (b) Decentralized. (C) Hybrid.

different servers, while the servers themselves are interconnected, and maintain a distributed database of peers and contents in order to be able to address requests that cannot be served locally.

The introduction of multiple servers reduces the impact of a single server failure, thus strengthening the reliability of the system. Concurrently with this multiple server scheme, eDonkey introduced file fragmentation, allowing peers to simultaneously download different segments of a content from several different sources, thereby increasing the throughput.

In a decentralized architecture (depicted in Figure 11.7b), such as the one used by Gnutella, each peer is connected to a small number of logical “neighbors” that it identifies through control messages. When a peer desires to acquire a content, it issues a request to its neighbors that, if they cannot provide the content themselves, transfer the request to their own neighbors. If a peer receives a request for a content

it can provide, it replies to the original requester and agrees with it on the terms of the transfer. The main advantage of this kind of architecture is the availability, due to the lack of a single point of failure that could affect the entire system. This architecture also favors anonymity, since no central database of all the connected peers is kept. The principal drawback is the signaling cost for the construction of the initial set of neighbors and the overhead due to the replicated requests, with a consequent reduction of throughput. Furthermore, in this architecture, a peer may not want to respond to the request even though it can provide the content, or it may refuse to forward the request it receives to its own neighbors. This problem, peers not participating their fair share of resources in a P2P system thus affecting the overall performance, is commonly referred to as free riding.

More recently, hybrid architectures have been proposed that include elements from both the centralized and decentralized P2P architectures (Figure 11.7c). Instead of using a centralized server, the system elects a subset of peers referred to as super peers. Super peers maintain a list of the peers assigned to them and the content they provide and are connected among each other in a Peer-to-Peer fashion, while regular peers connect to a super peer in a client/server fashion. In order to improve the availability, some architectures allow also regular peers to connect to more super peers at once. The actual transfer of the content is still carried on autonomously by the requesting peer and the serving peer, only the search of the content is (partially) centralized.

In general, the favorable properties that a P2P system should show are ease of search, low delay, low overhead, anonymity of peers, resilience to peer departures, and immunity to free riding. A great deal of research efforts has been invested in the development of more efficient algorithms capable of guaranteeing these properties. Furthermore, from a less technical perspective, P2P is still subject to legal, social, and moral debate. Several actors of the entertainment industry claim that P2P networks, allowing people worldwide to share files and data, encourage illegal transfers of copyright-protected contents, which in some countries has resulted in restrictions of P2P software usage.

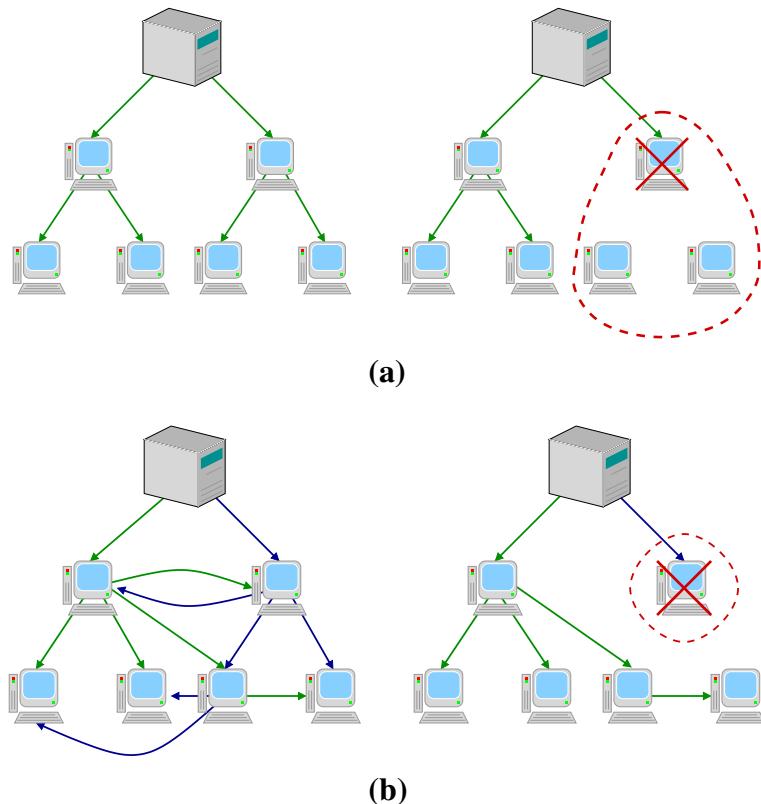
The development of media content distribution over P2P networks has encountered a few setbacks. First, most P2P systems suffer from free riding, which can limit the large-scale distribution of the content [35]. Second, in P2P networks there is the implicit assumption that a peer is connected for a long period of time, due to the traditional use in file sharing; conversely, in video streaming connection times can be small, with a high departures rate of peers. Along with high arrival rates, this situation is referred to as churn, a characteristic that P2P systems must be resilient to [36]. Finally, a video stream may require a high bitrate, so the bandwidth of users (both in upload and download) must be adequate, which is not always the case.

#### 5.11.3.4 Cooperative networking

To overcome the limitations of P2P distribution in the context of video streaming, the concept of cooperative networking (CoopNet) has been introduced. This new paradigm incorporates aspects of the CDN model into P2P distribution networks [32]. In the case of on-demand video services, the system initially operates in a purely client/server mode, with the server handling the transfer for a limited number of users. When more clients request the video stream and the server becomes overloaded, the server indicates to the requesting client the address of other clients that already have the desired video available in their cache memory, so that the service of the new client is provided in a P2P fashion. This scheme presents some analogies with the hybrid P2P architecture, but in CoopNet it is the content

itself that is provided by the central server, rather than the mere information on which nodes the content is located at. In the case of live streaming, clients do not obtain their video content from the cache memory of their peers, but they form instead logical structures, referred to as overlay networks, that provide application-layer multicast. Usually, the peers organize themselves in a tree structure rooted at the server; a newly connected client may connect directly to the video server, if it is not overloaded, or receive instructions to join the distribution tree and receive the content through the nodes of the tree.

Two major differences distinguish CoopNet from P2P networks. First, the presence of a central content server in CoopNet, which largely simplifies the content search. Second, while traditional P2P platforms are designed for file sharing applications and assume long connection times of the peers, in a CoopNet the system has to handle a much higher churn, being able to create, maintain, and repair quickly the overlay network in case of peer arrival and departure. This is extremely difficult to handle if the overlay consists in a single tree, as a single node failure would disconnect all the subtree rooted in that node, as depicted in Figure 11.8a. In order to cope with this problem, CoopNet systems



**FIGURE 11.8**

Examples of cooperative network overlay. (a) Single tree overlay. (b) Multi-tree overlay.

often use a redundant overlay, built as the superposition of multiple distribution trees, referred to as a multi-tree. This kind of structure is usually coupled with Multiple Description Coding (discussed in detail in [Section 5.11.7](#)), in order to efficiently divide the video content in descriptions, each one sent independently on a different tree. The multi-tree is organized in such a way that a peer cannot be directly connected to another peer in more than one tree, thus, if a peer disconnects, at most one description per node is affected, as depicted in [Figure 11.8b](#).

## 5.11.4 Streaming on wireless networks

In a wireless network terminals are not connected by cables of any kind, facing a significantly lower capacity and a much higher expected packet loss rate than in wired networks. This is due to several factors: the physical transmission on the channel is less reliable, due to power attenuation, fading, shadowing and multi-user interference, and other physical issues resulting in a time- and location-varying channel conditions. Also, mobile terminals rely on battery power, which is a scarce resource, and are far less reliable than Internet servers, routers, and clients.

The devices in a wireless network may be set up to either communicate indirectly through a central access point, or directly among each other. The former setup is referred to as *infrastructure network* and is typically employed to provide Internet access to mobile devices, having the access point acting as a gateway.

The latter setup, called mobile ad-hoc network, or MANET, is a dynamic, self-organizing, infrastructure-less network of mobile devices, interconnected by wireless links in a mesh topology [[37](#)].

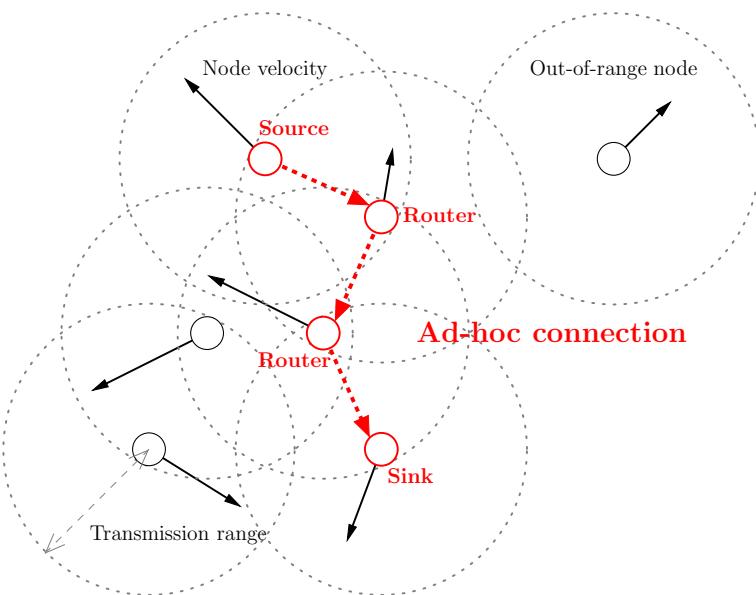
The devices, or nodes, of the ad-hoc network move freely and independently in all directions; consequently, the channel conditions of the links and the links themselves may change frequently. Also, individual nodes may connect and disconnect asynchronously, *i.e.*, without previous notice, and therefore their presence cannot be relied upon.

Each node of the network may initiate a communication with any other node. In order to deliver these data packets, other nodes in the network—even if unrelated to the communication—may have to participate in the transmission by forwarding the traffic. In other words, all nodes may at any time be source, sink, and router for a data stream. A simple example of mobile ad-hoc network is depicted in [Figure 11.9](#).

Ad-hoc networks may also be connected to the larger Internet, usually through a gateway node, connected via a wired connection, operating as an Internet connection sharing device. However, an ad-hoc network more commonly operates by itself, in order to support a specific application involving the nodes composing it. In fact, the Latin phrase *ad hoc* literally means “for this,” and in this context it is understood as “for this purpose.” This paradigm is also defined as application-driven networking.

A practically relevant case of application-driven networking is a network created by a group of people that use wireless computing to accomplish a collaborative task, which Feeney et al. call spontaneous network [[38](#)]. A spontaneous network reflects the fact that the nodes have chosen to cooperate for some purpose, and can therefore be leveraged into providing full cooperation to the network initialization and management. Another characteristic of spontaneous networks is to have a limited extension in both space and time, and that their population—although in principle unpredictable and dynamic—is expected to present relatively infrequent and minor changes over its lifetime [[38](#)].

MANETs offer a set of properties—flexibility, ease of deployment, robustness, *etc.*—that makes them applicable in environments without pre-existing communication infrastructure and where deploying it

**FIGURE 11.9**

A simple example of mobile ad-hoc network: nodes have velocities (black solid arrows) and a multi-hop ad-hoc transmission established (red dashed arrows). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this book.)

would be too expensive, too long, or simply not feasible. Common examples include students participating in an interactive lecture, business associates sharing information in a meeting, soldiers relaying situational awareness on the battlefield, or emergency disaster-relief personnel coordinating efforts after a hurricane or earthquake [39, 40]. Due to this possibility to create a network without infrastructure, ad-hoc networking has been defined as “the art of networking without a network” [37].

Since in ad-hoc networks nodes have to operate as routers, the pivotal challenge is to define a protocol that controls the policy by which nodes decide how to route the packets. The main issue is that nodes are unfamiliar with the topology of their networks, and thus have to announce their presence and in turn listen for announcements broadcasted by their neighbors, learning about nearby nodes and how to reach them, and possibly announcing to others how to reach them. Such protocols have responsibility of routing packets delivery—including routing through intermediate routers—which is normally implemented at network level (ISO OSI Model Level 3 [41, Ch. 5]), *i.e.*, on top of the Data Link Layer. Specifically, routing protocols are built on top of Data Link’s upper sub-layer, known as Medium Access Control, or MAC.

#### 5.11.4.1 Routing protocols for ad-hoc networks

The most commonly used MAC Layer for wireless networks is by far the standard IEEE 802.11 and its subsequent amendments, collectively known as 802.11x [42]. The growth of laptops, and more recently smartphones and tablets, implementing 802.11x wireless networking has made MANETs a popular

research topic for the last few years. Many works exist evaluating ad-hoc routing protocols and their performances assuming varying mobility models within a bounded space, usually with all nodes within a few hops of each other. Protocols performance is usually measured in terms of packet loss rate, protocol overhead, end-to-end delay, network throughput, *etc.* [39, 43–45]. In the following we shall give a short overview, in chronological order, of the most used ad-hoc routing protocols, summarized in Table 11.1.

The Destination-Sequenced Distance Vector Algorithm (DSDV) is a hop-by-hop distance vector routing protocol in which each node periodically broadcasts a routing update [46]. It is based on the classical Bellman-Ford routing algorithm for single-source shortest paths in a weighted directed graph [47]. DSDV is a table-driven algorithm; each entry in the routing table contains a sequence number, which is even if a direct link is present and odd otherwise. The number is generated by the destination, and the sender needs to communicate the next update with this number. Routing information is distributed between nodes by sending full dumps infrequently and smaller incremental updates more frequently. However, the amount of information that needs to be exchanged makes DSDV unsuitable for networks where topology changes occur frequently. The main contribution of this algorithm was to solve the routing loop problem that traditional distance vector protocols (such as the Bellman-Ford itself) did not.

The Dynamic Source Routing Algorithm, or DSR, is a reactive source-routing protocol for wireless mesh networks [48]. A reactive routing protocol establishes a route to a destination only on-demand; in contrast, most routing protocols of the Internet are proactive, *i.e.*, they find routing paths independently of the usage of the paths. DSR follows the source routing paradigm, a technique in which the sender of a packet—instead of relying on the routing tables at each intermediate node—can partially or completely specify the route that the packet should take through the network, based on the packet’s destination. Source routing requires that the complete, ordered address list of the nodes the packet will traverse is carried in the header of the packet itself. In large networks, this may result in high overhead; however, the accumulated path information can be cached by intermediate nodes, so that the learned paths can be used to route their own packets. Information about the best path to one’s destination is gathered in a route discovery phase, based on flooding. The main benefit of source routing is that intermediate nodes need not maintain up-to-date route information, since the packets themselves already contain all the routing decisions. Moreover, the on-demand nature of the protocol eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols. However, since the routing is decided at the source and never updated along the path, a source with out-of-date information (due to a change in topology between discovery phases) can lead to inconsistent routes.

The Temporally-Ordered Routing Algorithm (TORA) is a highly adaptive, loop-free, distributed routing algorithm that aims at achieving a high degree of scalability using a “flat” (*i.e.*, non-hierarchical) approach [49]. It is based on the classical “link reversal” routing algorithm for loop-free routes in networks with frequently changing topology [50]. TORA builds and maintains a directed acyclic graph rooted at each destination, localizing algorithmic reaction to topological changes as much as possible, *i.e.*, avoiding far-reaching control messages. TORA is particularly suited to operate in highly dynamic mobile networking environments as it is a self-stabilizing protocol, *i.e.*, it is self-healing in the face of nodes or links failures. However, in order to achieve this, TORA does not use a shortest path solution, which is unusual for routing algorithms of this type. The key contribution of TORA is the localization of control messages to a very small set of nodes near the occurrence of a topological change.

The Ad-Hoc On-Demand Distance Vector, or AODV, is, as the name indicates, a reactive distance vector routing protocol [51]. It is essentially based on a combination of both DSDV and DSR. It is similar

**Table 11.1** Comparative Overview of Routing Protocols for Ad-Hoc Networks

Protocol	Proposed by	Year	Type	Based on	Advantages	Drawbacks
DSDV	Perkins and Bhagwat	1994	Proactive	Distance vectors	First to solve the routing loop problem	Not suitable for highly dynamic networks
DSR	Broch, Johnson, and Maltz	1994	Reactive	Source routing	No routing information stored at intermediate nodes	Not suitable for highly dynamic networks
TORA	Park	1997	Proactive	Source rooted trees	Reduced overhead. Localized reactions	Does not provide the shortest path
AODV	Perkins, Belding-Royer, and Das	1999	Reactive	Hybrid	Capable of both unicast and multicast. Low connection setup delay	Stale entries in intermediate nodes can lead to inconsistent routes
OLSR	Jacquet and Muhlethaler et al.	2001	Proactive	Multi-point relay	Stable and fast. Reduced control overhead	Unable to track fast moving nodes

to DSR in that it borrows the basic mechanism of forming routes on-demand when a transmitting node requests one, but integrates it with the hop-by-hop routing, sequence numbers, and periodic beacons from DSDV in order to assure loop avoidance. Capable of both unicast and multicast routing, AODV provides a synthesis of the benefits of other protocols mentioned above and is nowadays, together with DSR, one of the most commonly used.

The Optimized Link-State Routing, or OLSR, is a proactive table based routing [52]. The key idea of OLSR is to define, for each node, a set of *multi-point relays* (MPRs). Rather than declaring all its links, a node only declares the set of its MPRs, minimizing the control traffic. OLSR provides two main functionalities: Neighbor Discovery, through which each node detects the neighbors with which it has a direct link, and Topology Dissemination, with which the node exchange topology control messages with its randomly selected MPRs and maintains topological information about the network. Being a proactive protocol, OLSR performs well in terms of latency to find a route. Moreover, the use of MPRs allows to significantly reduce the number of control messages and retransmission. However, in networks where nodes show a fast mobility, OLSR is unable to track rapid changes of topology, which results in a dramatic drop in performance. To overcome this limitation, Benzaid et al. have proposed an extension of OLSR, called Fast-OLSR, designed to meet the need for fast mobility [53]. Based on the OLSR protocol, Badis, Munaretto et al. have proposed an innovative link-state QoS routing protocol for ad-hoc networks known as QOLSR [54–56]. This protocol uses bandwidth and delay heuristic measures in order to select the MPRs, this improving the optimal path in terms of QoS requirements.

Most of the available routing protocols do not operate efficiently with networks of more than a few hundred nodes. Due to the growing importance of ad-hoc paradigm in applications involving a large population of mobile nodes, a great attention has been recently devoted to ad-hoc routing protocols with satisfactory scalability requirements, such as the Augmented Tree-based Routing Algorithm, or ATR [57]. This algorithm utilities an augmented tree-based address space structure and hierarchical multi-path routing in order to solve the scalability problem and to gain good resilience against node failure and link congestion.

#### 5.11.4.2 Multi-hop broadcast in ad-hoc networks

Most routing protocols make large use of broadcast in order to spread the information about possible routes; considering its wide use as a building block for other protocols, there is therefore a natural interest in a technique that efficiently delivers a packet from one node to all other nodes in the network.

Furthermore, broadcast also has an interest *per se* since many applications, among which video streaming, require one-to-many communication. The most commonly used techniques to perform multi-hop broadcast, also referred to as flooding, can be classified in four categories [58]: Simple Flooding, *Probabilistic Flooding*, *Area Based Flooding*, and *Neighbor-Knowledge Based Flooding*.

Simple Flooding is a widely used scheme for delivering a message to every part of a MANET in order to diffuse link-state information for routing purposes. The algorithm starts with the source broadcasting a message to all its neighbors; each of those neighbors, in their turn, re-broadcast the message exactly once, after a random assessment delay (RAD) [59]. Duplicate messages are dropped without re-broadcasting. The algorithm continues until, eventually, all reachable nodes have received the message. The popularity of this technique has several reasons. First, it is very simple to implement even in devices with little computational power on-board. Also, it is usually very fast, since the message

is sent through all possible paths, thus it is sent through the shortest path as well. However, even though this technique may be very effective for diffusing link-state information, which has a low bitrate and loose timing constraints, it is dangerously wasteful in terms of bandwidth in the case of general purpose broadcast, *e.g.*, video streaming: the increased load placed upon the network affects both the total available bandwidth and other nodes' resources (*e.g.*, battery power). Also, since more collisions occur, even in a connected network it is not guaranteed to reach all nodes. Furthermore, it is quite sensible to the choice of the RAD. Too short a RAD will affect reliability (because of collisions); too long, it will affect latency (because of the increased average delay). Simple Flooding has been proposed as a scheme to achieve multi-hop broadcast and multicast in highly dynamic wireless networks [60], and there exists an IETF Internet Draft proposing the use of Simple Flooding for broadcasting and multicasting in ad-hoc networks characterized by low node densities and high mobility [61].

*Probabilistic Flooding* is a technique similar to Simple Flooding proposed to mitigate the aforementioned problems. In this scheme, nodes only re-broadcast with a predetermined probability. It does not require major modifications from Simple Flooding and in dense networks, where multiple nodes share similar transmission ranges, randomly having some nodes not re-broadcasting saves node and network resources without harming reliability. Collision probability is also reduced, but in sparse networks, where there is much less shared coverage, nodes could not receive all the messages, unless the probability is high. This technique is also unfit for broadcasting a video stream, since there is no control over the path between a node and the source, hence on the QoE of the receivers. When the retransmission probability is set to 100%, this scheme is identical to Simple Flooding [62].

A different approach is provided by *Area Based Flooding* [58], wherein the nodes only consider the area covered by a transmission, rather than whether a node exists within that area. If nodes are somehow able to estimate their relative positions, using the signal strength or a GPS, they can infer how much additional area can be covered by their retransmission. Even though they perform generally better than both Simple and *Probabilistic Flooding*, Area-Based methods incur in a disproportionately increased number of retransmissions per packet as the number of source packets or the number of nodes in the network increase.

In all the broadcast schemes discussed so far, the nodes of the network retain barely any information on the topology to decide whether or not to retransmit a packet. *Neighbor-Knowledge Based Flooding* [58] differs in this respect, as each node is required to keep record of its neighbors, usually within one or two hops of radius, and the identity of the nodes it has received packets from. This allows it to determine whether it would reach additional nodes by retransmitting the packet. Neighbor-Knowledge Based methods generally outperform all the other schemes here described, and closely approach the theoretical bound. The main drawbacks of these schemes are the higher memory and computational resources required to the mobile nodes, and the need to keep the neighbors' tables up-to-date. If the network changes very dynamically, this could require a significant overhead.

#### 5.11.4.3 Video streaming over ad-hoc networks

Video streaming in a wireless environments is a challenging task, which can be successfully accomplished only if the coding scheme used to compress the content is both efficient, in terms of rate-distortion, and network-friendly [63]. The H.264/AVC standard and its extensions (see [Section 5.11.1](#)) offer both of these characteristics, and provide a set of tools that allow to adapt the encoding to the

transmission of the coded video data over any kind of network, including wireless [64]. These tools include, but are not limited to, a flexible framework that allows the implementation of RD-optimized packet scheduling strategies, and error resilience tools for communication over lossy networks.

One possible solution for video transmission over a multi-hop wireless network is using the scalability features offered by the SVC extension of H.264/AVC. However, since in scalable coding the different layers present hierarchical dependencies, a suitable strategy of unequal error protection (UEP) has to be provided. More details about UEP and the technique used to enable it will be provided in [Section 5.11.5](#).

Majumda et al. have proposed such a strategy for both unicast and multicast transmission on wireless networks that minimizes an assigned rate-distortion cost function, a solution that efficiently combines scalable coding with Forward Error Correction [65]. However, their solution only applies to video multicast in the last hop, *i.e.*, there is only one hop between the source and the destination.

A solution for the multi-hop scenario has been proposed by Cataldi et al., who suggested to use scalable video coding combined with rate-less codes, also known as *fountain codes* [66]. Rate-less codes are such that a potentially limitless sequence of codewords can be generated from a given set of source symbols, and the original source symbols can be recovered from any subset of codewords of size equal to or only slightly larger than the number of source symbols [67].

In particular, they propose a low-complexity class of rate-less codes based on a sliding-window approach applied to Raptor codes. Raptor codes are the first known class of rate-less codes with linear encoding and decoding time [68]. These codes have several properties useful for video applications, and provide better performance than classical rate-less codes. This system has shown a good end-to-end quality and robustness toward fluctuations in the packet loss rate.

Another approach that delivers from the need to provide unequal error protection is Multiple Description Coding (see [Section 5.11.7](#)). This approach (which had already been proven very efficient by Gogate et al. [69] in the transmission of still images over ad-hoc networks) has been investigated by Lin and Mao et al. for video delivery [10, 70]. In their comparative analysis of multi-path video streaming techniques over MANETs, they proved that the MDC approach is better suited for ad-hoc environments than scalable video coding. However, in order for MDC to be effective, a suitable routing protocol has to ensure that multiple (ideally, edge-disjoint) paths exist between the source and each receiver.

A class of routing schemes that ensure this property are the multi-tree construction protocols introduced in [Section 5.11.3](#). The basic idea of these schemes is to send each description over a different tree, constructed to be at least partially disjoint with each other so as to increase robustness to loss and other transmission degradations.

In order to prove this, Wei et al. have proposed two different multi-tree construction schemes, with the explicit goal of routing an MD coded video stream [71]. The first scheme constructs two disjoint trees in a serial and distributed fashion, and achieves reasonable tree connectivity while maintaining the trees completely disjoint. The second one, designed to reduce the control overhead and the construction delay, is a parallel protocol that generates nearly-disjoint trees. Their results show that the achieved video quality for either scheme is significantly higher than that of single tree multicast, while it presents only a slightly higher control overhead and a similar forwarding efficiency.

While Wei et al., and most of the other works on video transmission over ad-hoc networks existing at the time of their proposal, focused only on network-oriented metrics, such as throughput, delay and packet loss rate, Wu et al. [72] proposed an application-centric approach for real-time video transmission in ad-hoc networks, where expected video distortion is adopted as the routing metric. This is done using

a quality-driven cross-layer optimization that enhances the flexibility and robustness of the routing by jointly optimizing the path selection and the video coding, under a given video playback delay constraint. Their results, both theoretical and experimental, demonstrate that this approach outperforms the existing network-centric routing approaches. Although highly interesting, the approach of Wu et al. applies to directed acyclic graphs rather than real networks, and network-related problems are out of their scope.

Another important challenge of multi-hop video delivery in ad-hoc network, especially relevant when multi-path routing is used, is to limit the congestion generated by the transmitting nodes. This is a complicate issue, as the transmission policy of a single node streaming a video flow may impact the overall network congestion.

In order to cope with this problem, wireless streaming system may be integrated with a Congestion-Distortion Optimization (CoDiO) framework, an approach that has already proven viable in the design of cross-layer protocols for video streaming on MANETs [73].

The idea of Congestion-Distortion Optimization in video streaming, as opposed to the traditional rate-distortion optimization, was introduced to model the effects that self-inflicted network congestion has on video quality [74, 75]. This model was introduced considering a wired scenario wherein each node is connected to the video source by a succession of high-bandwidth shared links and terminating with a bottleneck on the last hop; it was later adapted for the case of unicast streaming over mobile ad-hoc networks by Zhu et al., who proposed a routing algorithm that seeks to minimize the congestion by optimally distributing the traffic over multiple paths [76]. They developed a model that captures the impact on the overall video quality of both the encoding process and the packet losses due to network congestion. Their model has proven to be able to capture the influence of these different parameters, and can be used to predict the end-to-end rate-distortion performance of a multi-hop video delivery system. However, this model provides neither an on-line estimation of the model parameters (*i.e.*, the network conditions), nor a viable extension for multicast streaming.

### 5.11.5 Error control, detection, concealment

Transmission errors are an inherent characteristic of any communication system. When video streaming is concerned, errors can have catastrophic effects, since the dependencies among the elements of the stream are such that an error on a single bit can affect a large amount of following data.

For these reasons, the problem of error control and concealment for video communications has gathered the attention of the research community for a long time [9, 77–79], and, as a consequence, many solutions exist. An effective classification of these methods, proposed in [9], is based on three categories. In a first one, referred to as *forward error concealment*, the encoder has the major role, since it injects in the video stream some amount of redundancy, which helps in recovering from errors. A second category is the one of *error concealment by postprocessing*: the decoder tries to replace corrupted data with estimation from surrounding information. Third, there are methods based on *communication and cooperation between encoder and decoder*.

#### 5.11.5.1 Error detection

These methods rely on the ability of *detecting* errors. For example data unit numbering for packets or pictures allows to detect missing units. More sophisticated solutions are based on the data semantics: the

decoded image content is examined and an error is declared when one finds abnormal characteristics, such as vertical variations greater than a given threshold [80,81] or too large differences across a macroblock (MB) borders [82]. At a lower semantic level, there are methods relying on the lossless coding structure: non-assigned codewords for an incomplete variable length code (VLC), or an incorrect number of decoded coefficients are interpreted as the result of a transmission error. Loss of synchronization in an arithmetic decoding can be used to the same scope.

Finally, the most reliable methods for error detection are based on header information or FEC codes [83], but they demand an additional rate. However, they can be used jointly with content-based methods, so that the trade-off between redundancy and reliability can be fine-tuned.

### 5.11.5.2 Error concealing

A first, natural approach to *forward error protection* consists in scalable coding followed by a prioritization of the obtained layers [84,85]. In this approach, the base layer, containing a rough version of the original video, is more protected from errors than the enhancement layers, which convey only details: this approach is often referred to as unequal error protection (UEP). These issues are covered in more detail in [Section 5.11.6](#).

Using SVC and UEP makes sense when the transport layer offers several logical channels with different error probabilities: this can be obtained using different channel codes or different physical channels. However, a lossless delivery of the base layer is not possible or feasible. An alternative solution would be the use of Multiple Description Coding (MDC) [86]. The basic idea is to represent the encoded video as a set of independently decodable and mutually refinable substreams, and to send them through (logically) independent channels. Any received stream allows decoding the video; the more streams are received, the better the final quality is. This resilience demands however an increased redundancy in the encoded stream; we refer the reader to [Section 5.11.7](#) for more details on MDC techniques and trade-offs.

The authors of [9,87] regard both scalable video coding and Multiple Description Coding as forms of joint source and channel coding (JSCC), since some hypotheses on the channel behavior are assumed in the design of the source coder: the possibility of UEP is assumed for SVC; the existence of independent path is at the basis of the MDC. However, in the scientific literature, the term JSCC refers mainly to techniques that take into account the channel characteristics at a lower level, namely in the design of the quantizer, the entropy coder, the FEC, and the modulation scheme. For example, it was noticed that coarse quantization is more robust in presence of high levels of noise [88]. Early studies were devoted to the joint minimization of error and quantization noise [89–91]. More recent JSCC solutions include the work by Cheung and Zakhor [92], who model the relationship among source code bitrate, channel code bitrate, and resulting distortion, for a wavelet-based scalable video coder; other RD models were described in [93] (for a DCT-based system) and in [94] (for the use of H.264/AVC on error-prone networks).

Another relevant class of forward error concealing techniques goes under the name of robust waveform coding (RWC) [9]. RWC techniques assume a single channel, and try to add redundancy to critical elements of the stream. For example, the MPEG-2 syntax allows sending motion vectors (MVs) even for macroblocks within an I frame [95]. If there is no error MVs are useless, but if a MB is corrupted, its MV can be inferred from neighbors, using then motion compensation to fill in the missing pixels. The *redundant picture* (RP) tool, introduced in H.264/AVC [13] (but already present in H.263 under the

name of sync picture [96]), is another example of RWC. For any given picture, further representations (the RPs) behind the primary one can be inserted into the H.264 stream. The RPs have typically a lower quality (and rate) than the primary picture: if the latter is lost, one can hopefully recover the missing image from the RP, which otherwise is obviously useless. Other possible usages of the RP are described in [97–100]. RP is supported in H.264/SVC as well; its properties can be signaled to the decoder, such that the latter is aware if the RP can completely replace the primary representation for the different kinds of prediction (temporal, spatial, inter-layer) [79].

Error resiliency can be increased by restricting the prediction domain, thus reducing the error propagation. The introduction of slices in H.263 and H.264/AVC [13, 101] implements this idea. The constrained domain can affect both temporal and spatial prediction. A somewhat similar principle is behind the intra MB/picture refresh tool: intra macroblocks or pictures are inserted in the bitstream to stop error propagation. A similar idea is the one of isolated regions [102], that jointly limits temporal and spatial prediction on a region-of-interest basis. Flexible macroblock ordering (FMO) in H.264 extends the capability to define slices beyond the simple case of consecutive MBs in raster-scan order, providing thus a simple and standardized method to perform region-based coding [103]. This in turns allows error-resiliency features such as constrained prediction, but also MDC or UEP.

Another class of forward error concealing tools is related to the use of other syntax elements of standard bitstreams [79]. A similar idea is at the basis of the data partitioning tools: headers and important data such as Intra frames and motion vectors are separated from the residuals (and no longer interleaved on a MB-by-MB basis) in order to better protect the former [77]. The concept is similar to UEP, but it works at a lower (*i.e.*, syntax) level. Data partitioning has been introduced in H.263, MPEG-4 and H.264/AVC [13, 14, 101]. In H.264, Intra slices residual are further separated from other residuals [7].

Adding redundancy to the encoding process is not limited to the waveform coding stage: even the lossless coding can be made more resilient by using robust entropy coding (REC) methods. Errors in a VLC stream can affect the synchronization of the decoder; in order to mitigate this problem a synchronization codeword (SC) can be included into the stream. In the MPEG-4 standard [14], SCs may also separate motion information from texture allowing to use the former if the latter is damaged. Another REC tool in H.263 and MPEG-4 [77] is the reversible variable length coding (RVLC): it allows decoding the bitstream backward, starting from a SC. Thus, when a bit between two SCs is wrong, instead of losing everything from the wrong bit until the second SC, some of that information can be recovered. This comes at the cost of a slightly decreased compression efficiency with respect to ordinary VLC.

Finally, at the encoder side error resiliency can be improved after the video coding process, by using classical channel codes on the compressed stream: Reed-Solomon codes [83], LDPC [104] and turbo codes [105] are common solutions. Transport level solutions, (based for example on interleaving in order to disperse errors in many packets) are also possible: for details, refer the reader to Section 5.11.2.

Now we switch to methods for *error concealing by postprocessing*, based on the decoder operation only. These techniques have the advantage of not requiring extra bandwidth nor imposing a delay. We consider the case when one or more MBs in a picture are corrupted: the coding mode, the MVs and/or the residual could be missing. However, often spatial or temporal neighbors are available: the *error concealment* methods try to estimate from them the missing information for the current MB.

A first class of solutions are based on temporal correlation. For example, in [106] the lost areas are replaced using blocks of the reference frame compensated by the average MV of the neighboring blocks. In [107] the MV and the low frequencies DCT coefficients are encoded in the base layer of a

scalable stream. If the enhancement layer is lost, instead of simply setting to zero the high frequencies, they are copied from the motion-compensated reference. These techniques are based on the hypothesis that MV information is available in the bitstream: when this is not true, one can resort to MV recovery techniques. For example, in [108], four candidates for MV recovery were considered: zero MV, the MV of the collocated block in the reference frame, and the average or the median of the MVs from the spatially adjacent blocks. The latter typically gave the best results. In [109] the choice among these candidates is performed by minimizing the border discontinuities. Subsequently, more sophisticated variations of this method were proposed: Chen et al. [110] use a partial differential equation algorithm to reduce the border discontinuities. Finally, temporal correlation can even be used when a whole picture is missing: *e.g.*, it can be estimated by using motion-compensated temporal interpolation techniques [111, 112], originally developed in the context of distributed video coding.

A second class of solutions take advantage of spatial regularity of images: the missing information is estimated by maximizing the smoothness of the resulting image [113]. This method was extended in order to take into account temporal smoothness [114] and to reduce the blurring artifacts [115]. An alternative to the smoothness constraint is to use the projection onto convex sets (POCS) [116] or to interpolate the missing information in the frequency domain [117]. POCS methods give better results than maximal smoothness ones, but they are computationally intensive and do not exploit temporal redundancy. Using motion-compensated interpolation and motion vector recovery is much simpler, but sometimes can produce noticeable artifacts.

A last family of error concealing techniques relies on the possibility of *communication and cooperation between the encoder and the decoder*. At source code level, the decoder can ask for the next frame to be encoded in Intra mode whenever an error is detected, in order to stop propagation [9]. To reduce the increase in bitrate, only some part of the next image may need to be encoded in Intra mode, due to the limited range of MVs [118]. A more sophisticated scheme is based on the identification of all blocks affected by a packet loss and error propagation via motion-compensation [106]: only these areas will be requested in Intra mode. A better solution would be to use the switching slice types provided by H.264/AVC [119]: in the case of a packet loss, the client signals the lost frame to the server, which in response sends an SP-frame, based only on the reference pictures correctly received by the client. Recently, a couple of cooperative error concealing tools have been integrated into of H.264/SVC. The first, called quality layer integrity check signaling [79], allows the decoder to detect missing layers with a CRC code and to inform the encoder; the latter can decide to encode the following pictures only using layers available to the decoder as references. The second is called temporal level zero index signaling [79], and can be used by the decoder to detect loss in the temporal hierarchy, allowing then to send a feedback to the sender (*e.g.*, a retransmission request). Finally, when the packet loss rate is communicated to the sender by the receiver, the former can perform a loss-aware rate-distortion optimization (LA-RDO) of the encoding process [120]. LA-RDO techniques can be used jointly with scalable video coding, for enhanced flexibility [121].

### 5.11.5.3 Error control

At the transport level, the most obvious solution for error control is retransmission of lost data. However this approach requires memory buffers and involves delays which could affect the streaming service. Therefore, retransmission strategies must always be coupled with suitable delay and buffer

management [9]. Other solutions include to change the rate of the FEC code, the GOP structure or the transmission schedule according to reported channel condition [122, 123].

### 5.11.6 Scalable video coding

Scalable video coding is a coding scheme intended to allow a single encoding of a video sequence, but enabling decoding from partial streams depending on the specific requirements of the application [124, Ch. 5].

This feature is crucial to provide an heterogeneous service without the need to re-encode the stream. For instance, the same flow can be used to offer the video content to users with different requirements in terms of resolution, frame rate, video quality, *etc.* The different demands of the users, which may depend on their preferences or on the erratic channel conditions, may be satisfied by simply extracting, from the encoded bitstream, the partial stream most suited to their needs.

In scalable video coding, the video data are divided into a base layer and one or more enhancement layers, and is therefore also referred to as *layered video coding*. Discarding appropriate layers from the scalable bitstream, different subbitstreams can be extracted to meet heterogeneous preferences and requirements in terms of quality of service (QoS).

The layers are designed in a hierarchical way, which makes progressive reconstruction at increasingly higher quality possible. The enhancement layers are useless unless the base layer and all the enhancement layers of lower detail are received. The video client can negotiate with the video server the number of layers it is interested in, according to its quality demands and resource availability.

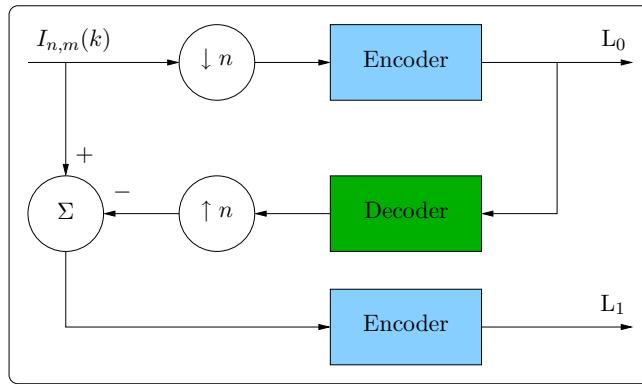
The most important scalable features are spatial resolution, temporal resolution, and quality (or SNR).

#### 5.11.6.1 Temporal scalability

Temporal scalability is a technique to encode a video sequence in a set of layers providing an increasing temporal resolution, with the perceivable effect of a progressive increase of the frame rate. It can be easily achieved by skipping some of the frames of the bitstream, with the *caveat* that the dependencies among layers should reflect the dependencies among frames dictated by the temporal prediction. For instance, with reference to the GOP structure depicted in Figure 11.3, a base layer can be constructed by the even frames and one enhancement layer by the odd ones. This type of GOP structure, known as hierarchical, is particularly interesting in this respect, as it allows the construction of as many layers as prediction levels there exist.

#### 5.11.6.2 Spatial scalability

In spatial scalability, the layers provide an increasing spatial resolution. A scheme to generate a two-layer spatially scalable video is presented in Figure 11.10. The base layer  $L_0$  is obtained by subsampling the original sequence  $I_{n,m}(k)$  of a factor  $N$ , then encoding it with a classical (*i.e.*, non-scalable) coder. An enhancement layer  $L_1$  is obtained up-sampling (*i.e.*, interpolating) the reconstructed lower layer and using it as a prediction of  $I_{n,m}(k)$ . The residual sequence is then in turn encoded with the classical coder. More layers can be obtained using a higher subsampling factor for the base layer, and progressively up-sampling, up to the original resolution.

**FIGURE 11.10**

Examples of a spatially scalable coding scheme for two layers.

### 5.11.6.3 Quality scalability

In SNR scalability, also referred to as bitrate scalability or quality scalability, the layers provide the same spatial and temporal resolution, but an increasing quantization accuracy. One possible technique to achieve SNR scalability is to encode the transform coefficients of the original sequence using a coarse quantizer  $Q_0$  to obtain the base layer, then de-quantize the quantized coefficients and use them as prediction of  $I_{n,m}(k)$ . The residue is encoded with a finer quantizer  $Q_1$  in order to obtain the enhancement layer  $L_1$ . More layers can be obtained using a progressively finer quantization. An alternative technique is to directly use an embedded coding of the transform coefficients, *e.g.*, using bit-plane coding.

In layered coding, an important problem is related to the generation of the temporal prediction for the current block. If the encoder uses information from all the layers, a mismatch can be generated between the references used at the encoder and those used at the decoder (drift effect) if the latter only receives the base layer. However, a prediction based on the base layer only will always be worse than it could have been if all the enhancement layers were allowed in the prediction.

This means that scalability combined with hybrid video coding comes at the price of a less performing encoder in terms of rate-distortion performance. In other words, at each rate, a scalable stream provides in general a lower video quality than the corresponding non-scalable video technique.

Since this inefficiency is mainly due to the prediction loop, which causes a drift problem whenever incomplete information is decoded, an alternative approach to the codec structure has been proposed, based on motion-compensated temporal filters instead of motion-compensated temporal prediction, in the framework of a multi-resolution transform, such as the Wavelet Transform [125–130].

---

## 5.11.7 Multiple Description Coding

Multiple Description Coding (MDC) is a coding framework allowing an improved immunity toward losses in unreliable networks, may losses be due to congestion or other perturbations, in case no feedback channel is available or retransmission delay is not tolerable [86]. Originally proposed at the beginning

of the 1970s for the speech signal, MDC has since been applied to other fields, such as image coding and video coding.

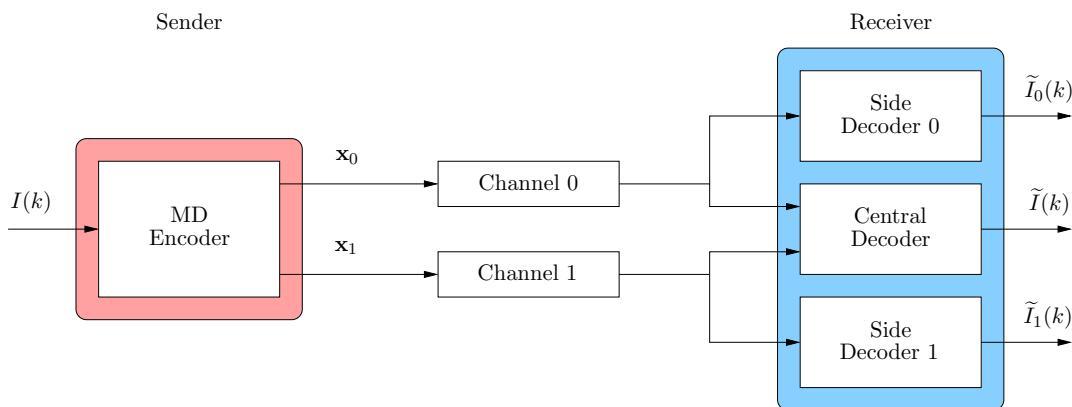
Goyal, among others, ascribe MDC to the class of joint source-channel coding techniques for erasure channels, since it takes simultaneously into account the possibility of losses and the encoding process, even though in general no explicit loss model is used in the design [131].

Whereas scalable video coding (see [Section 5.11.6](#)) is the state-of-the-art solution when preferential treatment can be performed on packets, *i.e.*, when more important packets can be provided with a better protection against failures, MDC handles the case of multiple independent transmission channels with roughly the same reliability.

In MDC, the source generates several versions of the same signal, called descriptions. Each description is independently decodable, but with a higher number of descriptions the quality of the reconstruction is improved. The main idea behind MDC is the independent transmission over independent channels of several representations of the same source signal. As we shall see, this imposes a trade-off between robustness and coding efficiency, in terms of compression ratio for a given quality. For the sake of clarity, we shall henceforth assume, without loss of generality, that two descriptions are generated.

A simple two-descriptions Multiple Description Video Coding system is represented in [Figure 11.11](#) (this scheme can be easily generalized in the case of more descriptions). A source video sequence  $I(k)$ , with frame index  $k \in \mathbb{N}$ , has to be transmitted to an end-user, having two independent lossy channels available. The *MD encoder* generates two compressed representations of the signal,  $x_0$  and  $x_1$ , referred to as descriptions of  $I(k)$ . Then, each description is sent over a different channel [132].

The two descriptions are independently decodable, *i.e.*, each one of them can be used to reconstruct a low-quality version of  $I$ . It is noteworthy that this is a crucial difference between MDC and SVC (introduced in [Section 5.11.6](#)): whereas in a scalable video coding technique there exist a strictly necessary base layer and a number of optional refinement layers, in MDC all the descriptions are independent of each other and can be used interchangeably. This is important when it is impractical or unfeasible to provide an unequal protection toward errors and losses to the base layer and the refinement



**FIGURE 11.11**

Scheme of a two-channels multiple description system.

layers. On the other hand, using MDC, the mere fact of sending the descriptions over different channels provides a degree of immunity to the stream.

At the receiver side, two different scenarios might occur. In the first scenario, only one description  $\mathbf{x}_d, d \in \{0, 1\}$ , is received correctly, while the other is affected by a loss on the channel. In that case, the decoding process is performed by a side decoder, which produces an approximated version of  $I$  based only on  $\mathbf{x}_d$ , here denoted as  $\tilde{I}_d$ . In the second scenario, both descriptions are correctly received. The decoding process is performed by the central decoder, which produces a reconstructed version of  $I$  based on both  $\mathbf{x}_0$  and  $\mathbf{x}_1$ , here denoted  $\tilde{I}$ . As a general rule,  $\tilde{I}$  has a lower distortion than both  $\tilde{I}_0$  and  $\tilde{I}_1$ . Notice that this scheme can be easily generalized in the case of more than two descriptions.

A traditional, *i.e.*, non-MDC encoding technique (in literature also referred to as *Single Description Coding*, or SDC), is normally optimized for rate-distortion efficiency, and the redundancy of the representation reduced by the encoding process. Conversely, any MDC technique is inherently affected by a certain degree of redundancy due to the correlation among the descriptions.

The RD characterization of an MD coded bitstream is a little more articulated than the one presented in [Section 5.11.1](#) for SD streams, as it has to take into account the possibility of either side decoder or the central decoder being used at the receiver. At any bitstream can be associated a quintuple  $(R_0, R_1, D_0, D_1, D)$ , where  $R_0$  and  $R_1$  are the rates of the two encoded descriptions  $\mathbf{x}_0$  and  $\mathbf{x}_1$ ,  $D_0$  and  $D_1$  are the distortions of  $\tilde{I}_0$  and  $\tilde{I}_1$  with respect to  $I$ , and  $D$  is the distortion of  $\tilde{I}$  with respect to  $I$  [133].

An important theoretical result of El Gamal and Cover states that for all achievable quintuples only when one or both side distortions are large, the central reconstruction can be very good; otherwise, there is a penalty in the central distortion [86].

This means that, in raw RD-performance, a SD technique usually outperforms an MD-technique if no losses occur, in the sense that, given the total rate of the descriptions, the quality of the reconstruction of an MD codec is lower than the one achievable with a SD codec at the same rate [133–135].

However, MDC becomes a viable tool whenever the stream has to be sent over a lossy channel: in this case, the introduction of a controlled redundancy in the MD-stream may be used to provide the end-user with an acceptable quality even if a large part of the stream is lost.

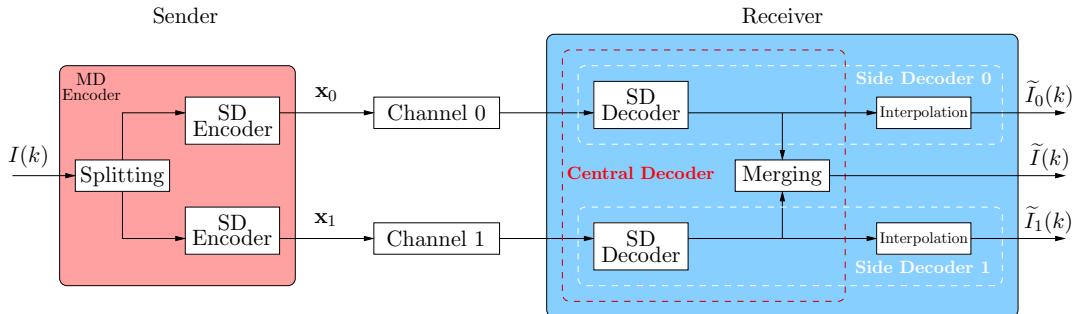
A central point in the design of any MDC technique is therefore to tune the degree of correlation among the descriptions: on one hand, redundancy in the representation is what grants this technique its loss resiliency; on the other hand, redundancy implies inefficiency in terms of rate-distortion optimization.

Historically, the first MDVC techniques to be proposed inherited from the MDC technique already proven efficient for still images [136, 137]. These techniques are commonly referred to as *intra-frame* or spatial MDVC techniques [132]. Other techniques also exploit the high degree of temporal correlation that video signals present.

Even though the latter approach, which we shall refer to as *inter-frame* or temporal MDC, shows good results, only a comparatively minor number of works have been proposed in this sense [138–140].

### 5.11.7.1 Multiple description channel splitting

The first MDC scheme to be introduced has been the channel splitting, originally meant for speech signal [141–143]. Channel splitting, depicted in [Figure 11.12](#), consists in the partition of the content of the original signal  $I(k)$ , usually achieved by polyphase subsampling, into a set  $\{I_0(k), I_1(k), \dots, I_{N-1}(k)\}$  of signals to be encoded independently in order to generate the descriptions. The reconstruction, in case

**FIGURE 11.12**

General scheme of a two-channel channel-splitting MD system.

some descriptions are missing, is generally performed through interpolation. The video signal can be split using temporal or spatial subsampling, generally producing balanced descriptions.

Temporal splitting for two descriptions consists in the separation of odd and even frames of the video sequence. The correlation between the substreams  $I_0(k)$  and  $I_1(k)$  depends on the degree of similarity of adjacent frames in the original sequence. When one description is missing, its samples can be approximated by temporally interpolating the other substream. The interpolation technique can be as easy as sample-wise sample-and-hold interpolation, which is equivalent to reducing the frame rate of the reproduction, or a more sophisticated technique such as *motion-compensated temporal interpolation* [111, 112]. When both descriptions are received, the two reconstructed substreams are merged to create the central reconstruction. The merging technique can be a simple interleaving of the frames of the two substreams, or a combination of received and interpolated sequence. This technique provides very high compression ratios, especially in regular motion video (such as video conferencing [132, 144]), it is easy to implement, and the descriptions can be encoded as standard-compliant bitstreams. This latter property is particularly important because even though existing standards can be used to provide MDC, there has not been a standard to explicitly address multiple description video coding, thus the ability to encode descriptions in a standard compliant way enables the practical adoption of the multiple description coding strategy.

Spatial splitting consists in partitioning each individual frame of the video sequence [144], *e.g.*, in even and odd rows. Spatial splitting, as temporal splitting, presents the advantage of providing good performance and of being easy to implement. In this case, the correlation among the descriptions is given by the spatial correlation among neighboring samples, so the quality of the side reconstructions depends on the regularity of the frames. However, spatial correlation and temporal correlation are very different in nature, and different interpolation techniques are used for side decoding.

### 5.11.7.2 Multiple description transform coding

The approaches discussed in the previous section are based on partitioning the signal in one of the domains it is defined on (time or space). The natural correlation between symbols in the source signal is exploited for reconstruction, *e.g.*, odd samples can be predicted from even samples, and vice-versa. When such techniques are employed, the degree of correlation among the descriptions depends only on

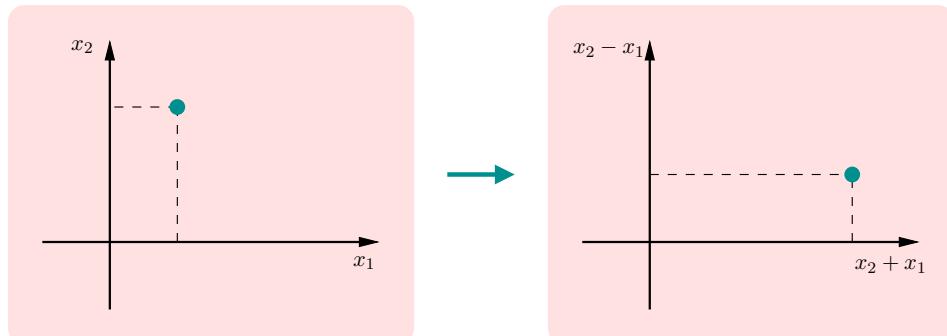
the statistics of the input signal. A considerably different approach to MD coding is to actively design a linear transform in order to finely control the degree of correlation between the descriptions of the source signal. This approach is referred to as MD transform coding. MD transform coding represents one of the most performing solutions for Multiple Description Coding [145–147]. It provides good energy compaction properties, resilience to additive noise and quantization, and great freedom to capture important signal characteristics. The correlation that remains after signal transformation can mitigate the effect of losses, since it offers the possibility to estimate the lost elements based on the received ones.

As discussed in [Section 5.11.1](#), in conventional SD video coding, spatial redundancy among samples is reduced via two-dimensional transform coding, so that the transform coefficients are less correlated and more compact. In contrast, *through a multiple description pairwise correlating transform* [147], the MD characteristic is achieved by introducing a known correlation between pairs of transform coefficients included in different descriptions. A key research issue is to control an appropriate type and amount of correlation.

Let us now consider the MD case, where the quantized versions of the transform coefficients are to be sent over two different channels. If one of the channel fails and one description is lost, being the coefficient poorly correlated, there would be no way to estimate the other.

To prevent this, the sequence of coefficients could be processed with a further transformation step, wherein a known correlation is introduced in order to allow an estimation of a missing description. An example of correlating transform is given in [Figure 11.13](#), where a signal  $\vec{x}$ , consisting of two independent Gaussian variables  $x_1$  and  $x_2$ , is transformed into  $\vec{y}$ , whose components are  $y_1 = x_1 + x_2$  and  $y_2 = x_2 - x_1$ , which can be shown to be optimal for independent Gaussian sources [148]. This transformation is such that the statistical dependencies between the components of  $\vec{y}$  allow from any one of them to estimate the original two components of  $\vec{x}$  to a certain accuracy, and the total estimation error for either component is the same. In practice, the cascade of a decorrelating and a pair-wise correlating transform is actually implemented as a single linear transform such that coefficients intended to the same description are internally decorrelated, and coefficients intended to different descriptions are correlated with each other[147].

The correlation between the descriptions improves the side decoder RD performance, as it is now possible to obtain an acceptable reconstruction of all coefficients given one description, but it also degrades the central performance. This method has been introduced for two descriptions in the context



**FIGURE 11.13**

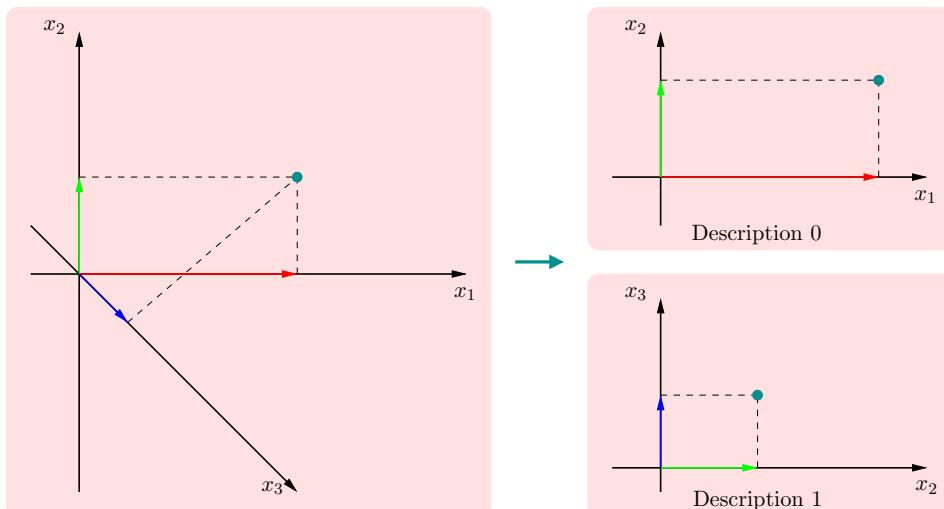
Example of correlating transform.

of image coding [148], and subsequently extend to more general mappings to produce an arbitrary number of descriptions [131, 146]. It is worth noticing that, whereas in SDC quantization is performed after the transformation, quantizing before applying a correlating transform has been shown to give the best performance [149, 147].

Another possibility to generate correlated representations of the same source signal via linear transform, is to project the signal onto an over-complete signal dictionary. For discrete signals, this means that the number of output coefficients will be larger than the number of input signal samples. Then, different subsets of coefficients can be included in different descriptions and sent over independent channels. Under some mild conditions, the redundant linear transform is a *frame expansion*, and the representation is known as *quantized frame expansion* (QFE) [150].

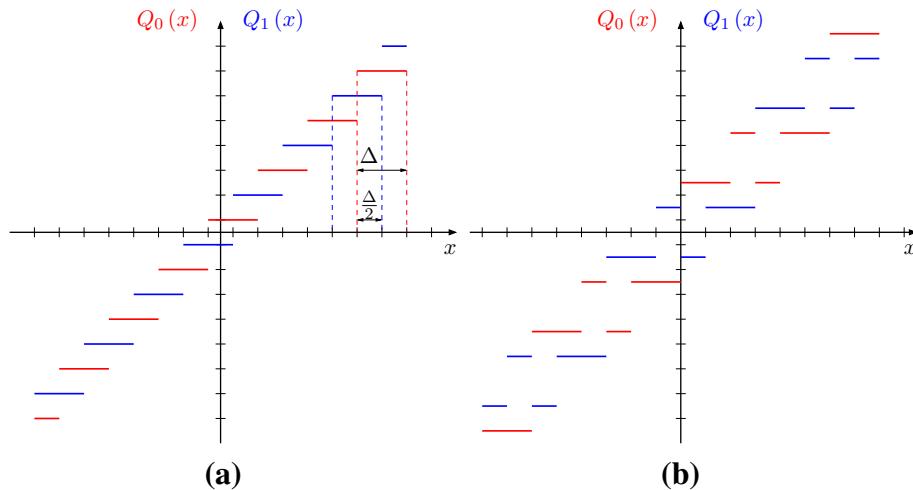
While redundant transform had already proven its robustness toward additive white noise and quantization, Goyal et al. proposed redundant transformation of the input signal as a way to achieve Multiple Description Coding in the context of transmission with an unpredictable number of losses [151–153]. Figure 11.14 shows a simple example of a redundant transform. In the original proposal, the source signal was estimated from the QFE coefficients as a least-squares problem. Later on, Chou et al. investigated how to provide a more efficient reconstruction of the original signal from any subset of quantized coefficients, thus enabling practical reconstructions from over-complete transforms, which had not been possible before [154]. Examples include windowed-DFT based schemes [155] and redundant wavelet based schemes [140, 156], the latter particularly interesting, as they are also inherently scalable.

It is worth noticing that the redundant wavelet based schemes also present the advantage of allowing scalability. An excellent survey on redundant wavelet schemes for Multiple Description Coding of video sequences can be found in [140]. Kovačević et al. investigated efficient oversampled filterbank implementations of redundant transform schemes for robust transmission over the Internet [157].



**FIGURE 11.14**

Example of redundant transform.

**FIGURE 11.15**

Examples of MD scalar quantizers. (a) Offset quantizer. (b) Non-convex quantizer.

More recently, the problem of optimal rate allocation for redundant transform MD schemes has been addressed, adapting the quantization of the transform coefficients to the importance of the basis functions, the redundancy in the representation, and the expected loss probability on the channel [158].

### 5.11.7.3 Multiple Description Quantization

In Multiple Description Quantization (MDQ), a scalar, vector, or entropy-constrained quantizer is designed to produce a number of descriptions, using a generalized Lloyd-like clustering algorithm that minimizes a Lagrangian cost function of rate and expected distortions [159]. MDQ has been one of the pioneering practical approaches to Multiple Description Coding [86]. In the example depicted in Figure 11.15a, the MDC character is achieved with two uniform quantizers of step  $\Delta$ , the second one being offset by half a quantization interval with respect to the first one. If one description is lost, the source signal is reconstructed from samples quantized with a step of  $\Delta$ ; if both descriptions are received, the resulting quantization step is  $\Delta/2$ .

In this scheme, if the side quantizers have a resolution of  $b$  bits, the central decoder has a resolution of approximately  $(b + 1)$  bits. In other words, when no losses occur, the system is using  $2b$  bits in order to have a resolution of  $b + 1$  bits, a very high redundancy that discourages the use of this technique unless the channel loss rates are very high and side reconstructions are very important. In order to overcome this drawback, Reudink invented several techniques with lower redundancy, such as the non-convex MD-quantizer [160], exemplified in Figure 11.15b. Later on, Vaishampayan et al. independently proposed a theoretical framework for designing MD scalar quantizers with fixed-rate quantization [161], subsequently extended to entropy-constrained quantization [162].

The MD-quantization techniques presented so far are completely agnostic with respect to the nature of the source signal (audio, image, video, etc.). More recently, MDVC schemes based on multiple

description scalar quantizer and the H.264/AVC standard have been proposed for reliable real-time video applications [163, 164].

Furthermore, approaches have been proposed for both video and still images that enable fine-grain scalability as well as robust coding of the input source, based on the joint use of the multi-resolution features of the DWT and an embedded multiple description scalar quantization [165, 166].

Multiple Description Video Coding, like scalable video coding (see Section 5.11.6), is known to suffer from drift effect when combined with motion-compensated prediction. That is, in presence of losses, the prediction signal available at the decoder may differ from the one used at the encoder, deteriorating the decoding process. In order to solve this problem, Crave et al. have proposed a robust scheme of Multiple Description Coding with side information that integrates an error control approach in the MDVC scheme applying Wyner-Ziv Coding (WZC) to every other frame of each description [167–169]. In WZC, a systematic error-correction code is applied on the video frame, possibly encoded using transform coding, but only the parity bits are transmitted [170, 171]. The decoder generates an estimation of the current frame, *e.g.*, by interpolating the adjacent frames. This estimation can be considered as a noisy version of the current frame, that the decoder corrects using the parity bits sent by the encoder.

### 5.11.8 Network coding

Since the beginning of the Internet, one of the key guidelines has been that, in a multi-hop communication, intermediate nodes do not alter the contents of packets: the routing problem was limited to selecting whether or not to send a copy of the received packet, and through which output link [41, 172, 173]. This has recently been challenged by the introduction of network coding (NC) [174].

Network coding is a generalization of routing wherein each packet sent over an output link of a node is a mixture of the packets received from the node's input links. Mixing ("coding") packets at intermediate nodes is beneficial to networking in several ways, increasing the throughput, reducing the delay, and granting error resilience—providing that a suitable decoding strategy is available at the receiver.

One aspect of networking where NC shows a clear advantage over traditional routing is multicast. One of the most celebrated results in network coding theory, the *Max-Flow-Min-Cut Theorem for Network Information Flows*, ensures that coding the packets within a network allows a source to multicast an information flow at a rate approaching the capacity of the smallest minimum cut between the source and any receiver, while the same result cannot be achieved through traditional routing [174]. The minimum cut between a source and a receiver is the smallest set of links that one must remove from the network in order to make the receiver unreachable from the source. This result has obviously created a great interest in network coding. Because of the wide range of applications that could benefit from it, such as video streaming, distributed information storage, and content delivery, many researchers have approached NC from a multitude of different points of view, such as graph theory, information theory, channel coding theory, and optimization theory.

The concept of network coding first appeared as a solution to the well-known *butterfly network problem* [174], depicted in Figure 11.16.

Given the topology in Figure 11.16, let us consider two sources,  $S_1$  and  $S_2$ , willing to deliver their respective messages  $x_1$  and  $x_2$  to two destinations  $D_1$  and  $D_2$ . All links have a capacity of one message per transmission. If intermediate nodes  $R_1$  and  $R_2$  can only forward the messages they receive, at

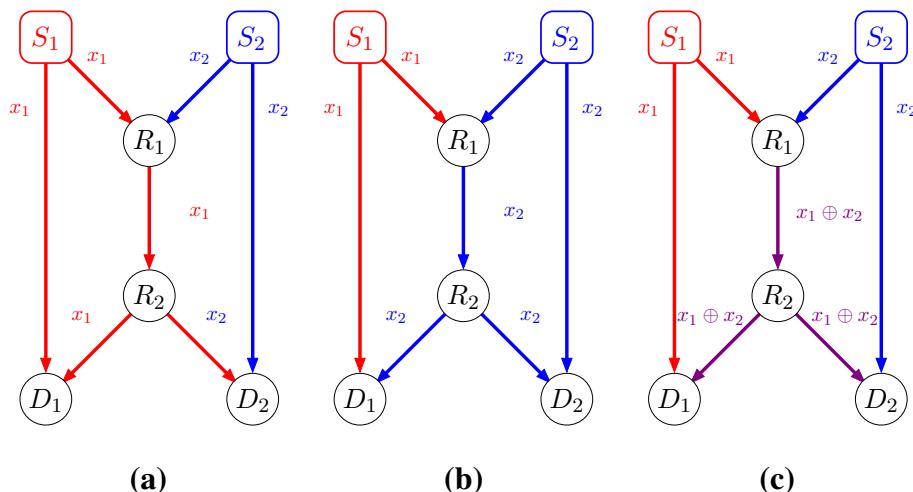


FIGURE 11.16

The butterfly network. Each edge represents a directed link with capacity of one message per transmission. Source nodes  $S_1$  and  $S_2$  want to transmit messages  $x_1$  and  $x_2$ , respectively, to both sink nodes,  $D_1$  and  $D_2$ . (a) Message  $x_1$  is sent over the bottleneck link. (b) Message  $x_2$  is sent over the bottleneck link. (c) A combination of  $x_1$  and  $x_2$  is sent over the bottleneck link.

every transmission they can either deliver  $x_1$  to both  $D_1$  and  $D_2$ , and  $x_2$  to  $D_2$  only (Figure 11.16a), or conversely  $x_2$  to both, but  $x_1$  to  $D_1$  only (Figure 11.16b). In other words, the link between  $R_1$  and  $R_2$  becomes a bottleneck. However, if node  $R_1$  is allowed to send a combination of  $x_1$  and  $x_2$ , e.g., the bit-wise exclusive-or, both receivers can obtain both messages with a single transmission per node, as shown in Figure 11.16c. As a result, while with the routing approach the two messages are delivered to all nodes with at least two successive transmission, using NC in the butterfly network, in a single transmission we can multicast two messages, which is exactly the capacity of the minimum cut between each source and each destination.

The exclusive-or can be generalized if we represent an unencoded information unit with an element of a finite field  $F_q$ , where  $q$  is the size of the field. A message of  $h$  units can be represented by a vector  $\mathbf{x} \in F_q^h$ . Using network coding, for any link  $e \in E$  of the network, the message is propagated as a *coded* symbol  $f_e(\mathbf{x}) \in F_q^h$ , with the encoding mechanism specified by a set of functions  $f_e(\cdot) \forall e \in E$ .

However, the theorem only gives a theoretical upper bound for the achievable multicast rate of the network, while it does not provide a constructive way to find the set of functions  $f_e(\cdot)$ . A great deal of effort has been therefore put into providing constructive solutions for different scenarios.

### 5.11.8.1 Linear network coding

A great leap in the field has been made when it has been proven that with a proper choice of  $q$ , the upper bound can be achieved using linear functions only [175]. In other words, each encoding function  $f_e(\mathbf{x})$

can be a linear combination of the type  $\mathbf{w}_e^\top \mathbf{x}$ , so that the network code is completely specified by the set of coding vectors  $\mathbf{w}_e$ .

Despite the indisputable theoretical value of these results, the design of the coding functions still requires a high degree of knowledge of the topology of the network, a centralized decision and a fixed assignment to the nodes of the network. All these requirements are hardly met on most real communication networks. In most communication networks, the structure, the topology, and the traffic demands may change quickly and drastically, while the information about those changes propagates with a certain delay. In wired networks, the edge capacities may vary due to changing traffic conditions and congestion. In wireless networks, they may vary in time due to fading channels, interference, and node mobility. Also, each change in the network would require the computation of a new set of optimal combination operations, with the associated computational cost. Therefore, subsequent work has investigated how to design algorithms capable of solving the multicast problem that could be implemented in practice.

In a real network, the coding functions can be assigned in a distributed fashion by performing Random Linear Network Coding (RLNC), *i.e.*, choosing the coefficients of the coding vectors independently and randomly over a suitable finite field [176–178]. In particular, it has been proven that the probability of a randomly chosen set of coefficients to ensure decodability at the receiver does not depend on the maximum multicast rate, and that this probability can be made arbitrarily close to one by working with a large enough field size  $q$ .

However, RLNC does not provide *per se* a method to transmit the coding and decoding functions, *i.e.*, the coding vectors, to the nodes where they take place. In order to fill this gap, Chou et al. presented the first practical approach to RLNC that takes into account the transmission of the coding vectors, which has become since then the most popular solution for RLNC [179]. Under the name of Practical Network Coding (PNC) it has shown promising results with respect to several problems in multimedia applications.

### 5.11.8.2 Practical Network Coding

In PNC, if the source message is composed of many units, as it is often the case in a real scenario, in order to reduce the number of coefficients in each coding vector, the data stream is divided into *generations*, each one consisting of  $k$  consecutive data packets of fixed size, with  $k$  much smaller than the size of the message. Only packets coming from the same generation can be combined at intermediate nodes. In those nodes, the received packets are stored in separate buffers per generation. When a sending opportunity occurs, a new packet is generated by combining, with random coefficients, all the packets in the current generation, and adding the coding coefficients in the packet's header. The operations needed to recover the source message can be found at the receiver by Gaussian elimination, as soon as  $k$  innovative (*i.e.*, linearly independent) coding vectors have been received.

The choice of the generation size is not trivial, as it involves a trade-off between decoding delay and rate overhead, *vs.* the decoding probability. On one hand, the decoding delay is negatively affected by a large generation size  $k$ , since a sink node has to wait for  $k$  innovative vectors in order to decode. Furthermore, the generation size also determines, together with the field size, the overhead due to the inclusion of coding vectors in the packet headers, which should be kept small, and particularly so in wireless networks, where packets are smaller and the overhead may become prohibitive. On the other hand, a large size of the generation affects positively the performance of the network coding in terms

of throughput, as more symbols are coded together (the Max-Flow Min-Cut Theorem assures that the maximum multicast flow is achieved only for  $h \rightarrow \infty$ ).

So far, we have discussed the topic of network coding from a graph-theoretical point of view. Little emphasis has been given on the several options that the ISO/OSI or the TCP/IP protocol stacks offer regarding the level wherein NC should be integrated [180]. Most of the solutions available perform network coding either at the Application Layer (OSI Layer 7), or at the Data-Link Layer (OSI Layer 2). Usually, the former approach is followed when coding is allowed only among packets belonging to the same multicast session, referred to as intra-session network coding. The latter usually allows combination of packets from different sessions (multicast or unicast), thus referred to as inter-session network coding.

Intra-session network coding, implemented at the application layer with little effort, has been proven beneficial for large-scale Peer-to-Peer (P2P) content distribution. In this kind of application, the source splits the content into small blocks, termed chunks, that are transmitted to each end-user in parallel. Once it has received a chunk, a user can trade it with anyone else interested in it against another chunk. Since in some architectures the source still distinguish itself for reliability and resources made available to the service, some prefer using the term cooperative rather than Peer-to-Peer, in the sense that the users are not considered peers, but clients that cooperate in order to alleviate the load of the sever [181]. The cooperative approach has been also applied to live multimedia streaming in large-scale networks (such as the Internet) [32, 182–185]. In this scenario, an uniform distribution of chunks is critical, as the chunk selection strategy has to take into account the play-out deadline of chunks as well [186] and an even larger benefit is to be expected by enabling network coding.

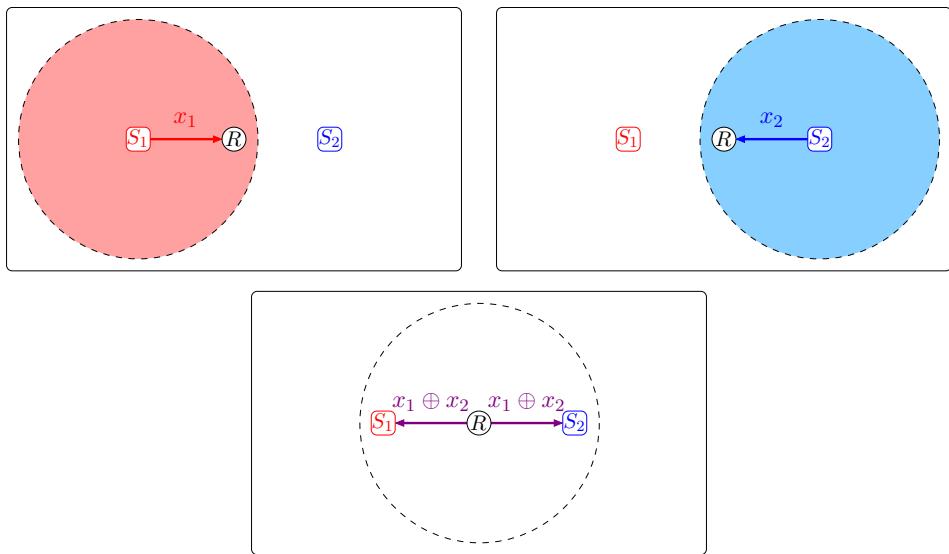
### 5.11.8.3 Network coding data dissemination

In self-organizing networks, such as MANETs, network coding can be used for wide dissemination of data as a substitute for flooding at the Data-Link layer. An example of the advantages of using NC in multi-hop broadcast over wireless ad-hoc networks is given in the simple scenario depicted in [Figure 11.17](#). In this scenario, two source nodes,  $S_1$  and  $S_2$  both want to broadcast a message, with node  $R$  as a relay. While using classical flooding node  $R$  would have to relay each message separately, using network coding it can relay the exclusive-or of the two, thus allowing both nodes  $S_1$  and  $S_2$  to decode each other's message. This is an important result, as energy efficiency (*i.e.*, the amount of battery energy consumed per transmitted bits) is a critical design parameter for mobile networks [187].

This is an example of inter-session network coding, in the sense that messages  $x_1$  and  $x_2$  may belong to different unicast or multicast sessions, oblivious to  $R$ . This approach has proven very efficient in the sense of maximization of the network throughput for both multiple unicast sessions [188–190], and multicast sessions [191–193].

However, network coding schemes merely aimed at throughput maximization are unfairly biased [194]. This happens as intermediate nodes, in order to decode their message, need to wait for the reception of the whole generation of encoded packets, some of which they may be not interested in. It has therefore been proposed to use a fair mixing strategy that takes into account the decoding delay of each destination. Basically, fair mixing consists in a form of dynamic intra-session network coding, where the session is not statically identified by the content, but rather by its end-point destination.

Furthermore, using inter-session network coding makes every packet dependent on other packets, so that even a single erasure or error, both extremely likely in unreliable environments such as wireless

**FIGURE 11.17**

Network coding for message exchange in multi-hop wireless network. The number of total transmissions is reduced from 4 to 3 if node  $R$  broadcasts a mixture of  $x_1$  and  $x_2$ , received from sources  $S_1$  and  $S_2$ , respectively.

networks, might affect the correct decoding of all packets. For this reason, several research efforts have been devoted to the design of robust strategies for NC, aimed at circumventing this limitation [195].

Given the good properties shown by network coding both in relaying video content in large-scale environments and in implementing a more efficient form of flooding in ad-hoc networks, it is interesting to study how network coding can be integrated in a framework for real time video delivery over wireless ad-hoc networks [196].

A possible strategy for video delivery in ad-hoc networks is the joint use of network coding and scalable video coding. As detailed in [Section 5.11.6](#), scalable video coding is a video coding paradigm that can account for the different requirements of quality of service and different network conditions by allowing the bitrate of the stream to dynamically adapt available bandwidth. This makes it suitable to be used in a video multicast scenario over heterogeneous networks such as MANETs. Since NC also has been shown to improve the throughput in such a network, it is interesting to evaluate how a joint design can further improve the performance in the network.

A drawback of using scalable coding, is that it poses a constraint in the order the layers are to be received, as higher layers can only be used if all previous layers have been received. Therefore, the delivery system must provide some form of unequal error protection in order to ensure that the lower layers are received with a higher probability than the higher ones.

In order to cope with this problem, a viable solution is to use Hierarchical Network Coding (HNC), a technique that allows the more important packets to be recovered with higher probability if only a small number of coded packets are received [197]. In this sense, HNC can be considered to provide unequal

error protection to the stream. Unequal error protection (UEP) means that parts of a stream that are more susceptible to errors causing a visible distortion (*e.g.*, in this case, the base layer) are provided with more protection (*i.e.*, more overhead) and vice-versa. To fine tune the probability of receiving a certain layer, a node can control the number of coded packets generated for that layer. With respect to RLNC, Hierarchical Network Coding allows the receivers to decode the most important packets earlier, but it suffers an overhead in decoding all less important packets. This is still a good results since, if there is not enough bandwidth, a receiver can be satisfied with the current number of reconstructed layers and instruct the sender to move on to the next chunk. A related technique is Expanding Window Network Coding (EWNC). The key idea of EWNC is to increase the size of the coding window for each new packet. In this sense, EWNC is a form of HNC where there exists a layer per packet. If in the buffer of the receivers the received coding vectors are kept in row echelon form, using Gaussian elimination, this method provides *instant decodability* of each packet if no losses occur.

Even though UEP scheme can mitigate the problem, the fact that layers have to be received in a predetermined order is still an inherent limitation of scalable video which Multiple Description Coding does not suffer, as discussed in [Section 5.11.7](#). Therefore, the system design when using NC jointly with MDC has to take into account fewer constraints. Thanks to the joint use of MDC and NC, users with high max-flow capacity will be able to satisfy their demands, *i.e.*, to receive more descriptions, whatever bottleneck may appear in the network at intermediate nodes. This result would not be possible using scalable coding, since in MDC any combinations of a given number of descriptions delivers (approximately) the same video quality, whereas missing a layer in a scalable coding would make it and any following layers useless [198].

The most recent trend, for multimedia applications especially, is to build protocols that are implemented as a cross-layer design between the application and network, so to be tailored to the specific media content to be delivered over a specific network type. This means that, when the transmitting live video streams, the network codes should maximize not only the network throughput, but also the video quality [199], taking into account the decodability of the code at the receivers as well as the deadlines of video packets and their contribution to the overall video quality. The first of this kind of techniques to be proposed, named Network Coding for Video (NCV), identifies at each sending opportunity a primary packet, *i.e.*, a packet that must be recovered by a given target node, and a possibly empty set of side packets, *i.e.*, packets that it is useful to include in the code in order to minimize the overall video distortion. The set of side packets could be empty depending on the state of the receivers' buffer as their inclusion in the code might make the recovery of the primary packet impossible at the target node. The selected packets are then coded together with RLNC and broadcast over the channel, along with information about their buffers. NCV also integrates a rate-distortion optimized packet scheduling framework, which allows to decide the current primary packet in a RD-optimal way, or to drop some packets altogether. Unfortunately, this method can achieve the global optimum for each transmission only if a perfect evaluation of the consequences, in terms of rate and total distortion, is available at each sending opportunity, which would require complete knowledge of the network topology and of the message exchanges among all nodes. However, simulation results also show that NCV performs well in practice with less message exchange and it can be considered an efficient heuristics to the RD-optimized version.

The NCV protocol gives little emphasis to the creation of the overlay network, and mainly focuses on the per-hop behavior. To obviate this problem, more recently, a video streaming solution based on a distributed receiver-driven overlay construction algorithm [200] has been proposed. Here, the approach

is reversed with respect to NCV, in the sense that the prioritization of packet is not performed as a open-loop optimization at the sender, but is based on the requests, made by the receivers, of packets of different layers. The sender then chooses the optimal coding strategy based on the requests, the priority of the packets, and the overall contribution to the local distortion. For overlay networks, a receiver-driven video streaming solution is proposed for video packets belonging to different priority classes. The problem of choosing the network coding strategy at every peer is formulated as an optimization problem of determining the rate allocation between the different packet classes such that the average distortion at the requesting peer is minimized.

---

## Conclusions

In this chapter, we presented an overview on the topic of multimedia streaming, with particular focus on video content transmission. We introduced the fundamental concepts of video coding, giving formal definitions of its goals and constraints, and an overview of the modern solutions available.

We then focused our attention on the standard transport protocols used to carry out the transfer of multimedia content in a streaming context. In particular, we presented a study of the MPEG-2 Transport Stream Protocol, commonly used for cable, satellite, and digital terrestrial TV, and the Real-Time Protocol, widely used for real-time communications over IP networks.

We analyzed the challenges involved in the setup of a streaming service in two very different scenarios. For wired networks, we discussed the three general approach nowadays used to provide a streaming service in a large-scale IP network: Content Distributions Networks, Application Layer Multicast, and Peer-to-Peer Networks. For wireless networks, and in particular for ad-hoc networks, we studied the routing methods used to sustain the quality of service required by a streaming service in highly dynamic and unreliable networks.

Furthermore, we provided an overview of the techniques used to reduce the impact that the inevitable transmission errors have on the quality of the stream. We discussed how this errors can be detected, and possibly corrected or otherwise concealed in order to improve the quality of the final reconstructed video.

We also presented two different coding schemes that can be used to provide the video stream with adaptivity to the network. In scalable video coding, the stream is divided in layers. The layers are organized in a hierachical way such that discarding the higher levels will reduce the bitrate needed for transmission still maintaining the stream decodable, at the price of a reduced video quality. In Multiple Description Coding, the substreams, or descriptions, are organized in a non-hierachical way such that any description is independently decodable and the video quality is increased with the number of decoded descriptions.

Finally we discussed the topic of network coding in the context of multimedia streaming. This is a transmission technique that can be applied both to wired and wireless communications and can provide a great benefit in terms of throughput and immunity to errors.

---

## Glossary

ALM	Application Layer Multicast
AODV	Ad-Hoc On-demand Distance Vector

ARP	address resolution protocol
ATR	Augmented Tree-based Routing
AVC	advanced video coding
CDN	Content Distribution Network
DCT	discrete cosine transform
DFT	discrete Fourier transform
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
DWT	discrete wavelet transform
EWNC	Expanding Window Network Coding
FEC	Forward Error Correction
FGS	fine granularity scalability
FMO	flexible macroblock ordering
GOP	group of pictures
GPS	global positioning system
HNC	Hierarchical Network Coding
IETF	Internet Engineering Task Force
IP	Internet protocol
JPEG	joint photographic experts group
MAC	Medium Access Control
MANET	mobile ad-hoc network
MC	motion compensation
MD	multiple description
MDC	Multiple Description Coding
MDCT	multiple description correlating transform
MDQ	Multiple Description Quantization
MDVC	Multiple Description Video Coding
MPEG	Moving Picture Experts Group
MV	motion vector
NC	network coding
NCV	Network Coding for Video
NTP	Network Time Protocol
OLSR	Optimized Link-State Routing
PNC	Practical Network Coding
QFE	quantized frame expansion
QoE	quality of experience
QOLSR	QoS-enhanced OLSR
QoS	quality of service
RAD	random assessment delay
RD	rate-distortion
RDO	rate-distortion optimization
RLNC	Random Linear Network Coding
RTP	real-time transport protocol

RTCP	real-time transport control protocol
SD	single description
SDC	Single Description Coding
SNR	signal-to-noise ratio
SVC	scalable video coding
TCP	transmission control protocol
TORA	Temporally-Ordered Routing Algorithm
UDP	user datagram protocol
UEP	unequal error protection
WZC	Wyner-Ziv Coding

---

## References

- [1] Michael McCandless, The MP3 revolution, *IEEE Intell. Syst.* 14 (3) (1999) 8–9.
- [2] Bob Ponce, The impact of MP3 and the future of digital entertainment products, *IEEE Commun. Mag.* 37 (9) (1999) 68–70.
- [3] G.K. Wallace, The jpeg still picture compression standard, *Commun. ACM* 34 (4) (1991) 30–44.
- [4] A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG 2000 still image compression standard, *IEEE Signal Process. Mag.* 18 (5) (2001) 36–58.
- [5] Hayder M. Radha, Mihaela Van der Schaar, Yingwei Chen, The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP, *IEEE Trans. Multimedia* 3 (1) (2001) 53–68.
- [6] P.A. Chou, Z. Miao, Rate-distortion optimized streaming of packetized media, *IEEE Trans. Multimedia* 8 (2) (2006) 390–404.
- [7] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, Ajay Luthra, Overview of the H.264/AVC video coding standard, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 560–576.
- [8] M. Kalman, E. Steinbach, B. Girod, Adaptive media playout for low-delay video streaming over error-prone channels, *IEEE Trans. Circ. Syst. Video Technol.* 14 (6) (2004) 841–851.
- [9] Yao Wang, Qin-Fan Zhu, Error control and concealment for video communication: a review, *Proc. IEEE* 86 (5) (1998) 974–997.
- [10] Shiwen Mao, Shunan Lin, S.S. Panwar, Yao Wang, E. Celebi, Video transport over ad-hoc networks: multi-stream coding with multipath transport, *IEEE J. Sel. Areas Commun.* 21 (10) (2003) 1721–1737.
- [11] Shiwen Mao, Xiaoling Cheng, Y. Thomas Hou, Hanif D. Sherali, Jeffrey H. Reed, On joint routing and server selection for MD video streaming in ad-hoc networks, *IEEE Trans. Wireless Commun.* 6 (1) (2007) 338–347.
- [12] Guy Côté, Berna Erol, Michel Gallant, Faouzi Kossentini, H.263+: video coding at low bit rates, *IEEE Trans. Circ. Syst. Video Technol.* 8 (7) (1998) 849–866.
- [13] Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496–10 (MPEG-4 AVC), Version 1: May 2003, Version 8: Consented in July 2007.
- [14] Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: April 1999, Version 3: May 2004.
- [15] ISO/IEC JTC1, Generic Coding of Moving Pictures and Associated Audio Information: Systems, ISO/IEC 13818-1, 1995.
- [16] Internet Engineering Task Force, Real-Time Transport Protocol, IETF RFC 3550, July 2003.
- [17] Internet Engineering Task Force, Network Time Protocol, IETF RFC 5905, June 2010.

- [18] S.E. Butner, M. Ghodoussi, Transforming a surgical robot for human telesurgery, *IEEE Trans. Rob. Autom.* 19 (5) (2003) 818–824.
- [19] G. Faria, J.A. Henriksson, E. Stare, P. Talmola, Dvb-h: digital broadcast services to handheld devices, *Proc. IEEE* 94 (1) (2006) 194–209.
- [20] S.B. Wicker, V.K. Bhargava, *Reed-Solomon Codes and Their Applications*, Wiley-IEEE Press, 1999.
- [21] D. Plets, W. Joseph, L. Verloock, E. Tanghe, L. Martens, E. Deventer, H. Gauderis, Influence of reception condition, mpe-fec rate and modulation scheme on performance of dvb-h, *IEEE Trans. Broadcast.* 54 (3) (2008) 590–598.
- [22] Internet Engineering Task Force, Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF), IETF RFC 4585, July 2006.
- [23] ETSI – DVB, Transport of MPEG-2 Based DVB Services Over IP Based Networks, ETSI TS 102 034, August 2009.
- [24] M. Kampmann, C. Plum, Stream switching for 3gpp pss compliant adaptive wireless video streaming, *Third IEEE Consumer Communications and Networking Conference 2006, CCNC 2006*, vol. 2, IEEE, 2006, pp. 954–958.
- [25] T. Stockhammer, Dynamic adaptive streaming over HTTP–: standards and design principles, in: *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ACM, 2011, pp. 133–144.
- [26] A. Zambelli, IIS smooth streaming technical overview, Microsoft Corporation (2009). <<http://www.microsoft.com/en-us/download/details.aspx?id=17678>>.
- [27] R. Pantos, W. May, HTTP Live Streaming, Internet Engineering Task Force (30 September 2011). <<http://tools.ietf.org/html/draft-pantos-http-live-streaming-07>>.
- [28] I. Sodagar, The MPEG-DASH standard for multimedia streaming over the internet, *IEEE MultiMedia* 18 (4) (2011) 62–67.
- [29] ISO-IEC JTC1, ISO Base Media File Format, ISO/IEC 14496-12.
- [30] R. Santos, C. Rocha, B. Rezende, A. Loureiro, Characterizing the YouTube Video-Sharing Community, White Paper, 2008.
- [31] B. Krishnamurthy, C. Wills, Y. Zhang, On the use and performance of content distribution networks, in: *Proceedings of ACM SigComm Workshop on Internet Measurement*, San Francisco, CA, USA, November 2001.
- [32] Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou, Kunwadee Sripanidkulchai, Distributing streaming media content using cooperative networking, in: *Proceedings of ACM SigComm International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Miami Beach, FL, USA, May 2002.
- [33] C. Diot, B.N. Levine, B. Lyles, H. Kassem, D. Balensiefen, Deployment issues for the IP multicast service and architecture, *IEEE Network* 14 (1) (2000) 78–88.
- [34] D. Jurca, J. Chakareski, J.-P. Wagner, P. Frossard, Enabling adaptive video streaming in P2P systems, *IEEE Commun. Mag.* 45 (6) (2007) 108–114.
- [35] Jiadi Yu, Minglu Li, Feng Hong, Guantao Xue, Free-riding analysis of BitTorrent-like peer-to-peer networks, in: *Proceedings of IEEE Asia-Pacific Conference on Services Computing*, Guangzhou, PRC, December 2006.
- [36] Daniel Stutzbach, Reza Rejaie, Characterizing Churn in Peer-to-Peer Networks Technical Report, University of Oregon, June 2005.
- [37] Magnus Frodigh, Per Johansson, Peter Larsson, Wireless ad-hoc networking: the art of networking without a network, *Ericsson Rev.* 4 (2000) 248–263.
- [38] L.M. Feeney, B. Ahlgren, A. Westerlund, Spontaneous networking: an application oriented approach to ad-hoc networking, *IEEE Commun. Mag.* 39 (6) (2001) 176–181.
- [39] Elizabeth M. Royer, Chai-Keong Toh, A review of current routing protocols for ad-hoc mobile wireless networks, *IEEE Pers. Commun.* 6 (2) (1999) 46–55.

- [40] M. Gerla, From battlefields to urban grids: new research challenges in ad-hoc wireless networks, Elsevier J. Pervasive Mobile Comput. 1 (1) (2005) 77–93.
- [41] Andrew S. Tanenbaum, Computer Networks, fourth ed., Prentice Hall, 2003.
- [42] R. Bruno, M. Conti, E. Gregori, Mesh networks: commodity multi-hop ad-hoc networks, IEEE Commun. Mag. 43 (3) (2005) 123–131.
- [43] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, Jorjeta Jetcheva, A performance comparison of multi-hop wireless ad-hoc network routing protocols, in: Proceedings of ACM International Conference on Mobile Computing and Networking, Dallas, TX, USA, October 1998.
- [44] Rüdiger Schollmeier, Ingo Gruber, Micheal Finkenzeller, Routing in mobile ad-hoc and peer-to-peer networks: a comparison, in: Proceedings of IEEE International Conference on Peer-to-Peer Computing, Linköping, Sweden, September 2002.
- [45] L. Hanzo, R. Tafazolli, Admission control schemes for 802.11-based multi-hop mobile ad-hoc networks: a survey, IEEE Commun. Surveys and Tutorials 11 (4) (2009) 78–108.
- [46] Charles E. Perkins, Pravin Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, ACM SigComm Comput. Commun. Rev. 24 (1994) 234–244.
- [47] Richard Bellman, On a routing problem, AMS Q. Appl. Math. 16 (1958) 87–90.
- [48] D.B. Johnson, D.A. Maltz, J. Broch, DSR: the dynamic source routing protocol for multi-hop wireless ad-hoc networks, in: T. Imielinski, H. Korth (Eds.), Ad-hoc Networking, vol. 5, Kluwer Academic Publishers, 2001, pp. 139–172.
- [49] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of IEEE International Conference on Computer Communications, Kobe, Japan, April 1997.
- [50] E. Gafni, D. Bertsekas, Distributed algorithms for generating loop-free routes in networks with frequently changing topology, IEEE Trans. Commun. 29 (1) (1981) 11–18.
- [51] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, February 1999.
- [52] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad-hoc networks, in: Proceedings of IEEE International Multitopic Conference, Lahore, Pakistan, December 2001.
- [53] M. Benzaid, P. Minet, K. Al Agha, Integrating fast mobility in the OLSR routing protocol, in: Proceedings of IEEE International Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.
- [54] Anelise Munaretto, Hakim Badis, Khaldoun Al Agha, Guy Pujolle, QoS for ad-hoc networking based on multiple metrics: bandwidth and delay, in: Proceedings of IEEE International Conference on Mobile and Wireless Communications Networks, Singapore, October 2003.
- [55] Hakim Badis, Anelise Munaretto, Khaldoun Al Agha, Guy Pujolle, Optimal path selection in a link state QoS routing protocol, in: Proceedings of IEEE Vehicular Technology Conference, Milan, Italy, May 2004.
- [56] Hakim Badis, Khaldoun Al Agha, Optimal path selection in ad-hoc networks based on multiple metrics: bandwidth and delay, in: Advances in Wireless Ad-Hoc and Sensor Networks, Springer, 2008, (Chapter 2).
- [57] M. Caleffi, G. Ferraiuolo, L. Paura, Augmented tree-based routing protocol for scalable ad-hoc networks, in: Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Pisa, Italy, October 2007.
- [58] Brad Williams, Tracy Camp, Comparison of broadcasting techniques for mobile ad-hoc networks, in: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland, June 2002.
- [59] A. Mohammed, M. Ould-Khaoua, L.M. Mackenzie, Improvement to efficient counter-based broadcast scheme through random assessment delay adaptation for MANETs, in: Proceedings of UKSim European Symposium on Computer Modeling and Simulation, Liverpool, England, UK, September 2008.

- [60] C. Ho, K. Obraczka, G. Tsudik, K. Viswanath, Flooding for reliable multicast in multi-hop ad-hoc networks, in: Proceedings of ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, WA, USA, August 1999.
- [61] J. Jetcheva, Y. Hu, D. Maltz, D. Johnson, A simple protocol for multicast and broadcast in mobile ad-hoc networks, IETF Internet Draft, July 2001.
- [62] Wei Peng, Xicheng Lu, AHBP: an efficient broadcast protocol for mobile ad-hoc networks, *J. Comput. Sci. Technol.* 16 (2001) 114–125.
- [63] T. Stockhammer, M.M. Hannuksela, T. Wiegand, H.264/AVC in wireless environments, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 657–673.
- [64] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G.J. Sullivan, Rate-constrained coder control and comparison of video coding standards, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 688–703.
- [65] A. Majumda, D.G. Sachs, I.V. Kozintsev, K. Ramchandran, M.M. Yeung, Multicast and unicast real-time video streaming over wireless LANs, *IEEE Trans. Circ. Syst. Video Technol.* 12 (6) (2002) 524–534.
- [66] P. Cataldi, M. Grangetto, T. Tillo, E. Magli, G. Olmo, Sliding-window raptor codes for efficient scalable wireless video broadcasting with unequal loss protection, *IEEE Trans. Image Process.* 19 (6) (2010) 1491–1503.
- [67] Michael Luby, LT codes, in: Proceedings of IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada, November 2002.
- [68] A. Shokrollahi, Raptor codes, *IEEE Trans. Inf. Theory* 52 (6) (2006) 2551–2567.
- [69] N. Gogate, Doo-Man Chung, S.S. Panwar, Yao Wang, Supporting image and video applications in a multi-hop radio environment using path diversity and multiple description coding, *IEEE Trans. Circ. Syst. Video Technol.* 12 (9) (2002) 777–792.
- [70] Shunan Lin, Yao Wang, Shiwen Mao, S. Panwar, Video transport over ad-hoc networks using multiple paths, in: Proceedings of IEEE International Symposium on Circuits and Systems, Scottsdale, AZ, USA, May 2002.
- [71] W. Wei, A. Zakhori, Multiple tree video multicast over wireless ad-hoc networks, *IEEE Trans. Circ. Syst. Video Technol.* 17 (1) (2007) 2–15.
- [72] Dalei Wu, Song Ci, Haohong Wang, A.K. Katsaggelos, Application-centric routing for video streaming over multihop wireless networks, *IEEE Trans. Circ. Syst. Video Technol.* 20 (12) (2010) 1721–1734.
- [73] Eric Setton, Taesang Yoo, Xiaoqing Zhu, Andrea Goldsmith, Bernd Girod, Cross-layer design of ad-hoc networks for real-time video streaming, *IEEE Trans. Wireless Commun.* 12 (4) (2005) 59–65.
- [74] I.V. Bajić, O. Tickoo, A. Balan, S. Kalyanaraman, J.W. Woods, Integrated end-to-end buffer management and congestion control for scalable video communications, in: Proceedings of IEEE International Conference on Image Processing, Barcelona, Spain, September 2003.
- [75] Eric Setton, Bernd Girod, Congestion-distortion optimized scheduling of video over a bottleneck link, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, Siena, Italy, September 2004.
- [76] Xiaoqing Zhu, Eric Setton, Bernd Girod, Congestion-distortion optimized video transmission over ad-hoc networks, *Signal Process. Image Commun.* Elsevier Sci. 20 (2005) 773–783, (Invited Paper).
- [77] Y. Wang, S. Wenger, J. Wen, A.K. Katsaggelos, Error resilient video coding techniques, *IEEE Signal Process. Mag.* 17 (4) (2000) 61–82.
- [78] S. Kumar, L. Xu, M.K. Mandal, S. Panchanathan, Error resiliency schemes in h. 264/avc standard, *J. Vis. Commun. Image Represent.* 17 (2) (2006) 425–450.
- [79] Y. Guo, Y. Chen, Y.K. Wang, H. Li, M.M. Hannuksela, M. Gabbouj, Error resilient coding and error concealment in scalable video coding, *IEEE Trans. Circ. Syst. Video Technol.* 19 (6) (2009) 781–795.
- [80] K.N. Ngan, R. Stelle, Enhancement of PCM and DPCM images corrupted by transmission errors, *IEEE Trans. Commun.* COM-30 (1982) 257–265.
- [81] K.M. Rose, A. Heiman, Enhancement of one-dimensional variable-length DPCM images corrupted by transmission errors, *IEEE Trans. Commun.* 37 (1989) 373–379.

- [82] W.-M. Lam, A. Reibman, An error concealment algorithm for images subject to channel errors, *IEEE Trans. Image Process.* 4 (1995) 533–542.
- [83] S. Lin, D.J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [84] M. Ghanbari, Two-layer coding of video signals for vbr networks, *IEEE J. Sel. Areas Commun.* 7 (5) (1989) 771–781.
- [85] A.E. Mohr, E.A. Riskin, R.E. Ladner, Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction, *IEEE J. Sel. Areas Commun.* 18 (6) (2000) 819–828.
- [86] Vivek K. Goyal, Multiple description coding: compression meets the network, *IEEE Signal Process. Mag.* 18 (5) (2001) 74–93.
- [87] Maria G. Martini, Matteo Mazzotti, Catherine Lamy-Bergot, Jyrki Huusko, Amon Peter, Content adaptive network aware joint optimization of wireless video transmission, *IEEE Commun. Mag.* 45 (1) (2007) 84–90.
- [88] L.J.J. Spilker Jr., *Digital Communications by Satellite*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [89] J. Kurtenbach, P.A. Wintz, Quantizing for noisy channels, *IEEE Trans. Commun. Technol.* CT-17 (1969) 291–302.
- [90] N. Farvardin, V. Vaishampayan, Optimal quantizer design for noisy channels: an approach to combined source-channel coding, *IEEE Trans. Inf. Theory* IT-38 (1987) 827–838.
- [91] J.W. Modestino, D.G. Daut, Optimal quantizer design for noisy channels: an approach to combined source-channel coding, *IEEE Trans. Commun. COM-7* (1979) 1644–1659.
- [92] G. Cheung, A. Zakhori, Bit allocation for joint source/channel coding of scalable video, *IEEE Trans. Image Process.* 9 (3) (2000) 340–356.
- [93] Z. He, J. Cai, C.W. Chen, Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding, *IEEE Trans. Circ. Syst. Video Technol.* 12 (6) (2002) 511–523.
- [94] Y. Zhang, W. Gao, Y. Lu, Q. Huang, D. Zhao, Joint source-channel rate-distortion optimization for h. 264 video coding over error-prone networks, *IEEE Trans. Multimedia* 9 (3) (2007) 445–454.
- [95] ISO/IEC JTC1, Generic Coding of Moving Pictures, ISO/IEC 13818-2, 1995.
- [96] S. Wenger, Video redundancy coding in H.263+, in: Proceedings of the International Workshop Audio-Visual Services Over Packet, Networks, September 1997.
- [97] Y.-K. Wang, M.M. Hannuksela, M. Gabbouj, Error resilient video coding using unequally protected key pictures, in: Proceedings of the International Workshop Very Low Bitrate Video (VLBV '03) 2003, Madrid, Spain, September 2003, pp. 290–297.
- [98] S. Rane, P. Baccichet, B. Girod, Modeling and optimization of a systematic lossy error protection system based on H.264/AVC redundant slices, in: Proceedings of Picture Coding Symposium (PCS '06), Beijing, China, April 2006.
- [99] I. Radulovic, Y.-K. Wang, S. Wenger, A. Hallapuro, M.M. Hannuksela, P. Frossard, Multiple description h.264 video coding with redundant pictures, in: Proceedings of Mobile Video Workshop, ACM Multimedia '07, Augsburg, Germany, September 2007, pp. 37–42.
- [100] C. Zhu, Y.K. Wang, M.M. Hannuksela, H. Li, Error resilient video coding using redundant pictures, *IEEE Trans. Circ. Syst. Video Technol.* 19 (1) (2009) 3–14.
- [101] International Telecommunication Union – Telecommunication Standardization Sector, Video coding for low bit rate communication, ITU-T Recommendation H.263, March 1996.
- [102] M.M. Hannuksela, Y.K. Wang, M. Gabbouj, Isolated regions in video coding, *IEEE Trans. Multimedia* 6 (2) (2004) 259–267.
- [103] Stephan Wenger, H.264/AVC over IP, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 645–656.
- [104] Y. Wang, S. Yu, X. Yang, Error robustness scheme for h. 264 based on ldpc code, in: Proceedings of 12th International Conference on Multi-Media Modelling 2006, IEEE, 2006, p. 4.

- [105] Z. Peng, Y.-F. Huang, D.J. Costello Jr., Turbo codes for image transmission-a joint channel and source decoding approach, *IEEE J. Sel. Areas Commun.* 18 (6) (2000) 868–879.
- [106] M. Wada, Selective recovery of video packet loss using error concealment, *IEEE J. Sel. Areas Commun.* 7 (5) (1989) 807–814.
- [107] L.H. Kieu, K.N. Ngan, Cell-loss concealment techniques for layered video codecs in an atm network, *IEEE Trans. Image Process.* 3 (5) (1994) 666–677.
- [108] P. Haskell, D. Messerschmitt, Resynchronization of motion compensated video affected by atm cell loss, *IEEE International Conference on Acoustics, Speech, and Signal Processing 1992, ICASSP-92, 1992*, vol. 3, IEEE, 1992, pp. 545–548.
- [109] W.M. Lam, A.R. Reibman, B. Liu, Recovery of lost or erroneously received motion vectors, *IEEE International Conference on Acoustics, Speech, and Signal Processing 1993, ICASSP-93, 1993*, vol. 5, IEEE, 1993, pp. 417–420.
- [110] Yan Chen, Yang Hu, O.C. Au, Houqiang Li, Chang Wen Chen, Video error concealment using spatio-temporal boundary matching and partial differential equation, *IEEE Trans. Multimedia* 10 (1) (2008) 2–15.
- [111] Claudio Greco, Marco Cagnazzo, Béatrice Pesquet-Popescu, H.264-based multiple description coding using motion compensated temporal interpolation, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, Saint-Malo, France, October 2010.
- [112] Claudio Greco, Giovanni Petrazzuoli, Marco Cagnazzo, Béatrice Pesquet-Popescu, An MDC-based video streaming architecture for mobile networks, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, Hangzhou, PRC, October 2011.
- [113] Y. Wang, Q.F. Zhu, L. Shaw, Maximally smooth image recovery in transform coding, *IEEE Trans. Commun.* 41 (10) (1993) 1544–1551.
- [114] Q.F. Zhu, Y. Wang, L. Shaw, Coding and cell-loss recovery in dct-based packet video, *IEEE Trans. Circ. Syst. Video Technol.* 3 (3) (1993) 248–258.
- [115] W. Zhu, Y. Wang, A comparison of smoothness measures for error concealment in transform coding, in: Proceedings of SPIE Conference Visual Communication and Image Processing, vol. 2, 1995, pp. 1205–1214.
- [116] H. Sun, W. Kwok, Concealment of damaged block transform coded images using projections onto convex sets, *IEEE Trans. Image Process.* 4 (4) (1995) 470–477.
- [117] S.S. Hemami, T.H.Y. Meng, Transform coded image reconstruction exploiting interblock correlation, *IEEE Trans. Image Process.* 4 (7) (1995) 1023–1027.
- [118] N. Mukawa, H. Kuroda, T. Matsuoka, An interframe coding system for video teleconferencing signal transmission at a 1.5 mbit/s rate, *IEEE Trans. Commun.* 32 (3) (1984) 280–287.
- [119] Marta Karczewicz, Ragip Kurceren, The SP- and SI-frames design for H.264/AVC, *IEEE Trans. Circ. Syst. Video Technol.* 13 (7) (2003) 637–644.
- [120] T. Stockhammer, D. Kontopodis, T. Wiegand, Rate-distortion optimization for jvt/h. 26l video coding in packet loss environment, in: Internet Packet Video, Workshop, 2002.
- [121] Y. Guo, Y.K. Wang, H. Li, Error resilient mode decision in scalable video coding, in: IEEE International Conference on Image Processing 2006, IEEE, 2006, pp. 2225–2228.
- [122] J. Chakareski, P. Frossard, Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources, *IEEE Trans. Multimedia* 8 (2) (2006) 207–218.
- [123] N. Tizon, B. Pesquet-Popescu, M. Cagnazzo, Adaptive video streaming with long term feedbacks, in: IEEE International Conference on Image Processing, Cairo, Egypt, 2009.
- [124] M. Van der Schaar, P.A. Chou, *Multimedia Over IP and Wireless Networks: Compression, Networking, and Systems*, Academic Press, 2007.
- [125] D. Taubman, A. Zakhor, Multirate 3-D subband coding of video, *IEEE Trans. Image Process.* 3 (5) (1994) 572–588.

- [126] J.-R. Ohm, Three dimensional subband coding with motion compensation, *IEEE Trans. Image Process.* 3 (5) (1994) 559–571.
- [127] S.J. Choi, J.W. Woods, Motion-compensated 3-D subband coding of video, *IEEE Trans. Image Process.* 8 (2) (1999) 155–167.
- [128] B. Pesquet-Popescu, V. Bottreau, Three-dimensional lifting schemes for motion compensated video compression, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2001, pp. 1793–1796.
- [129] D.S. Turaga, M. van der Schaaf, B. Pesquet-Popescu, Complexity scalable motion compensated wavelet video encoding, *IEEE Trans. Circ. Syst. Video Technol.* 15 (8) (2005) 982–993.
- [130] Thomas André, Marco Cagnazzo, Marc Antonini, Michel Barlaud, JPEG2000-compatible scalable scheme for wavelet-based video coding, *EURASIP J. Image Video Process.* 2007 (1) (2007) 9–19.
- [131] V.K. Goyal, J. Kovačević, Optimal multiple description transform coding of Gaussian vectors, in: Proceedings of Data Compression Conference, Snowbird, UT, USA, March 1998.
- [132] Y. Wang, A.R. Reibman, S. Lin, Multiple description coding for video delivery, *Proc. IEEE 93* (1) (2005) 57–70, (Invited Paper).
- [133] L. Ozarow, On a source-coding problem with two channels and three receivers, *Bell Syst. Tech. J.* 59 (10) (1980) 1909–1921.
- [134] A. El Gamal, T. Cover, Achievable rates for multiple descriptions, *IEEE Trans. Inf. Theory* 28 (6) (1982) 851–857.
- [135] R. Venkataramani, G. Kramer, V.K. Goyal, Multiple description coding with many channels, *IEEE Trans. Inf. Theory* 49 (9) (2003) 2106–2114.
- [136] A.R. Reibman, H. Jafarkhani, Yao Wang, M.T. Orchard, R. Puri, Multiple description coding for video using motion compensated prediction, in: Proceedings of IEEE Internantional Conference on Image Processing, Kobe, Japan, October 1999.
- [137] Wee Sun Lee, M.R. Pickering, M.R. Frater, J.F. Arnold, A robust codec for transmission of very low bit-rate video over channels with bursty errors, *IEEE Trans. Circ. Syst. Video Technol.* 10 (8) (2000) 1403–1412.
- [138] V.A. Vaishampayan, S. John, Balanced interframe multiple description video compression, in: Proceedings of IEEE Internantional Conference on Image Processing, Kobe, Japan, October 1999.
- [139] Yao Wang, Shunan Lin, Error-resilient video coding using multiple description motion compensation, *IEEE Trans. Circ. Syst. Video Technol.* 12 (6) (2002) 438–452.
- [140] Christophe Tillier, Claudia Teodora Petrisor, Béatrice Pesquet-Popescu, A motion-compensated overcomplete temporal decomposition for multiple description scalable video coding, *EURASIP J. Image Video Process.* 1 (2007) 1–12.
- [141] J.C. Candy, D. Gloge, D.J. Goodman, W.M. Hubbard, K. Ogawa, Protection by Diversity at Reduced Quality, Technical Report, Bell Labs Memo for Record (not archived), June 1978.
- [142] A. Gersho, The Channel Splitting Problem and Modulo-PCM Coding, Technical Report, Bell Labs Memo for Record (not archived), October 1979.
- [143] S.E. Miller, Fail-safe Transmission Without Standby Facilities Technical Report, in: TM80-136-2, Bell Labs, August 1980.
- [144] N. Franchi, M. Fumagalli, R. Lancini, S. Tubaro, Multiple description video coding for scalable and robust transmission over IP, *IEEE Trans. Circ. Syst. Video Technol.* 15 (3) (2005) 321–334.
- [145] V.K. Goyal, J. Kovačević, R. Arean, M. Vetterli, Multiple description transform coding of images, in: Proceedings of IEEE Internantional Conference on Image Processing, Chicago, IL, USA, October 1998.
- [146] V.K. Goyal, J. Kovačević, Generalized multiple description coding with correlating transforms, *IEEE Trans. Inf. Theory* 47 (6) (2001) 2199–2224.
- [147] Yao Wang, M.T. Orchard, V. Vaishampayan, A.R. Reibman, Multiple description coding using pairwise correlating transforms, *IEEE Trans. Image Process.* 10 (3) (2001) 351–366.

- [148] Yao Wang, M.T. Orchard, A.R. Reibman, Multiple description image coding for noisy channels by pairing transform coefficients, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, Princeton, NJ, USA, June 1997.
- [149] M.T. Orchard, Y. Wang, V. Vaishampayan, A.R. Reibman, Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms, in: Proceedings of IEEE International Conference on Image Processing, Washington, DC, USA, October 1997.
- [150] R.J. Duffin, A.C. Schaeffer, A class of non-harmonic Fourier series, *Trans. AMS* 72 (1952) 341–366.
- [151] V.K. Goyal, M. Vetterli, J. Kovačević, Multiple description transform coding: robustness to erasures using tight frame expansions, in: Proceedings of IEEE International Symposium on Information Theory, Cambridge, MA, USA, August 1998.
- [152] V.K. Goyal, M. Vetterli, N.T. Thao, Quantized overcomplete expansions in  $R^N$ : analysis, synthesis, and algorithms, *IEEE Trans. Inf. Theory* 44 (1) (1998) 16–31.
- [153] V.K. Goyal, J. Kovačević, M. Vetterli, Quantized frame expansions as source channel codes for erasure channels, in: Proceedings of Data Compression Conference, Snowbird, UT, USA, March 1999.
- [154] P.A. Chou, S. Mehrotra, A. Wang, Multiple description decoding of overcomplete expansions using projections onto convex sets, in: Proceedings of Data Compression Conference, Snowbird, UT, USA, March 1999.
- [155] R. Balan, Ingrid Daubechies, V. Vaishampayan, The analysis and design of windowed Fourier frame based multiple description source coding schemes, *IEEE Trans. Inf. Theory* 46 (7) (2000) 2491–2536.
- [156] Christophe Tillier, Béatrice Pesquet-Popescu, Mihaela Van der Schaar, Multiple descriptions scalable video coding, in: Proceedings of European Signal Processing Conference, Vienna, Austria, September 2004.
- [157] J. Kovačević, P.L. Dragotti, V.K. Goyal, Filter bank frame expansions with erasures, *IEEE Trans. Inf. Theory* 48 (6) (2002) 1439–1450, (Invited Paper).
- [158] I. Radulovic, Pascal Frossard, Multiple description image coding with redundant expansions and optimal quantization, in: Proceedings of IEEE Workshop on Multimedia Signal Processing, Crete, Greece, October 2007.
- [159] J.G. Proakis, M. Salehi, *Digital Communications*, McGraw-Hill, 2001.
- [160] D.O. Reudink, The Channel Splitting Problem with Interpolative Coders Technical Report, in: TM80-134-1, Bell Labs, October 1980.
- [161] V.A. Vaishampayan, Design of multiple description scalar quantizers, *IEEE Trans. Inf. Theory* 39 (3) (1993) 821–834.
- [162] V.A. Vaishampayan, J. Domaszewicz, Design of entropy constrained multiple description scalar quantizers, *IEEE Trans. Inf. Theory* 40 (1) (1994) 245–250.
- [163] T. Guionnet, C. Guillemot, S. Pateux, Embedded multiple description coding for progressive image transmission over unreliable channels, in: Proceedings of IEEE International Conference on Image Processing, Thessaloniki, Greece, October 2001.
- [164] O. Campana, R. Contiero, G.A. Mian, An H.264/AVC video coder based on a multiple description scalar quantizer, *IEEE Trans. Circ. Syst. Video Technol.* 18 (2) (2008) 268–272.
- [165] F. Verdicchio, A. Munteanu, A.I. Gavrilescu, J. Cornelis, P. Schelkens, Embedded multiple description coding of video, *IEEE Trans. Image Process.* 15 (10) (2006) 3114–3130.
- [166] Augustin I. Gavrilescu, Fabio Verdicchio, Adrian Munteanu, Ingrid Moerman, Jan Cornelis, Peter Schelkens, Scalable multiple description image coding based on embedded quantization, *EURASIP J. Image Video Process.* 1 (2007) 20–30.
- [167] Olivier Crave, Christine Guillemot, Béatrice Pesquet-Popescu, Christophe Tillier, Distributed temporal multiple description coding for robust video transmission, *EURASIP J. Wireless Commun. Networking* 2008 (2008) 1–12.

- [168] Olivier Crave, Christine Guillemot, Béatrice Pesquet-Popescu, Multiple description source coding with side information, in: Proceedings of European Signal Processing Conference, Lausanne, Switzerland, August 2008.
- [169] Olivier Crave, Béatrice Pesquet-Popescu, Christine Guillemot, Robust video coding based on multiple description scalar quantization with side information, *IEEE Trans. Circ. Syst. Video Technol.* 20 (6) (2010) 769–779.
- [170] A. Wyner, J. Ziv, The rate-distortion function for source coding with side information at the decoder, *IEEE Trans. Inf. Theory* 22 (1) (1976) 1–10.
- [171] Anne Aaron, Rui Zhang, Bernd Girod, Wyner-Ziv coding of motion video, in: Proceedings of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, November 2002.
- [172] D.E. Comer, D.L. Stevens, *Internetworking with TCP/IP*, Prentice-Hall, 1982.
- [173] James F. Kurose, Keith W. Ross, *Computer networking: a top-down approach featuring the Internet*, Pearson Education, 2004.
- [174] R. Ahlswede, Ning Cai, S.-Y.R. Li, R.W. Yeung, Network information flow, *IEEE Trans. Inf. Theory* 46 (4) (2000) 1204–1216.
- [175] S.-Y.R. Li, R.W. Yeung, Ning Cai, Linear network coding, *IEEE Trans. Inf. Theory* 49 (2) (2003) 371–381.
- [176] T. Ho, M. Médard, J. Shi, M. Effros, D.R. Karger, On randomized network coding, in: Proceedings of IEEE International Symposium on Information Theory, Kanagawa, Japan, June 2003.
- [177] T. Ho, R. Koetter, M. Médard, D.R. Karger, M. Effros, The benefits of coding over routing in a randomized setting, in: Proceedings of IEEE International Symposium on Information Theory, Kanagawa, Japan, June 2003.
- [178] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, B. Leong, A random linear network coding approach to multicast, *IEEE Trans. Inf. Theory* 52 (10) (2006) 4413–4430.
- [179] P.A. Chou, Y. Wu, K. Jain, Practical network coding, in: Proceedings of Allerton Conference on Communication Control and Computing, Monticello, IL, USA, October 2003.
- [180] E. Magli, P. Frossard, An overview of network coding for multimedia streaming, in: Proceedings of IEEE International Conference on Multimedia and Expo, New York, NY, USA, July 2009.
- [181] Venkata N. Padmanabhan, Kunwadee Sripanidkulchai, The case for cooperative networking, in: Proceedings of International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, March 2002.
- [182] Feng Wang, Yongqiang Xiong, Jiangchuan Liu, mTreebone: a hybrid tree/mesh overlay for application-layer live video multicast, in: Proceedings of IEEE International Conference on Distributed Computing Systems, Toronto, ON, Canada, June 2007.
- [183] Eric Setton, Pierpaolo Baccichet, Bernd Girod, Peer-to-peer live multicast: a video perspective, *Proc. IEEE* 96 (1) (2008) 25–38, (Invited Paper).
- [184] Feng Wang, Yongqiang Xiong, Jiangchuan Liu, mTreebone: a hybrid tree/mesh overlay for application-layer live video multicast, *IEEE Trans. Parallel Distrib. Syst.* 21 (3) (2010) 379–392.
- [185] Elie Gabriel Mora, Claudio Greco, Béatrice Pesquet-Popescu, Marco Cagnazzo, Joumana Farah, Cedar: an optimized network-aware solution for P2P video multicast, in: Proceedings of IEEE International Conference on Telecommunications, Jounieh, Lebanon, April 2012.
- [186] R. Cunningham, B. Biskupski, R. Meier, Managing peer-to-peer live streaming applications, in: Proceedings of Springer-Verlag International Conference on Distributed Applications and Interoperable Systems, Oslo, Norway, June 2008.
- [187] Dong Nguyen, Tuan Tran, Thinh Nguyen, B. Bose, Wireless broadcast using network coding, *IEEE Trans. Veh. Technol.* 58 (2) (2009) 914–925.
- [188] S. Katti, D. Katabi, W. Hu, H. Rahul, M. Medard, The importance of being opportunistic: practical network coding for wireless environments, in: Proceedings of Allerton Conference on Communication Control and Computing, Monticello, IL, USA, September 2005, (Invited Paper).

- [189] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, Jon Crowcroft, XORs in the air: practical wireless network coding, *ACM SigComm Comput. Commun. Rev.* 36 (4) (2006) 243–254.
- [190] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Médard, J. Crowcroft, XORs in the air: practical wireless network coding, *IEEE/ACM Trans. Networking* 16 (3) (2008) 497–510.
- [191] J. Widmer, C. Fragouli, J.-Y. Le Boudec, Low-complexity energy-efficient broadcasting in wireless ad-hoc networks using network coding, in: Proceedings of IEEE Workshop on Network Coding, Theory and Applications, Riva del Garda, Italy, April 2005.
- [192] C. Fragouli, J. Widmer, J.-Y. Le Boudec, A network coding approach to energy efficient broadcasting: from theory to practice, in: Proceedings of IEEE International Conference on Computer Communications, Barcelona, Spain, April 2006.
- [193] C. Fragouli, J. Widmer, J.-Y. Le Boudec, Efficient broadcasting using network coding, *IEEE/ACM Trans. Networking* 16 (2) (2008) 450–463.
- [194] Golnaz Karbaschi, Aline C. Viana, Steven Martin, Khaldoun Al Agha, On using network coding in multi hop wireless networks, in: Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Tokyo, Japan, September 2009.
- [195] Marco Di Renzo, Lana Iwaza, Michel Kieffer, Pierre Duhamel, Khaldoun Al Agha, Robust wireless network coding – An overview, in: Proceedings of ICST International Conference on Mobile Lightweight Wireless Systems, Barcelona, Spain, May 2010.
- [196] Christina Fragouli, Dina Katabi, Athina Markopoulou, Muriel Medard, Hariharan Rahul, Wireless network coding: opportunities and challenges, in: Proceedings of IEEE Military Communications Conference, Orlando, FL, USA, October 2007.
- [197] K. Nguyen, Thinh Nguyen, Sen ching Cheung, Peer-to-peer streaming with hierarchical network coding, in: Proceedings of IEEE International Conference on Multimedia and Expo, Beijing, PRC, July 2007.
- [198] A.K. Ramasubramonian, J.W. Woods, Multiple description coding and practical network coding for video multicast, *IEEE Signal Process. Lett.* 17 (3) (2010) 265–268.
- [199] Hulya Seferoglu, Athina Markopoulou, Opportunistic network coding for video streaming over wireless, in: Proceedings of IEEE Packet Video Conference, Lausanne, Switzerland, November 2007.
- [200] N. Thomas, J. Chakareski, P. Frossard, Prioritized distributed video delivery with randomized network coding, *IEEE Trans. Multimedia* 13 (4) (2011) 776–787.

# Multimedia Content-Based Visual Retrieval

# 12

Wengang Zhou<sup>\*</sup>, Houqiang Li<sup>\*</sup>, and Qi Tian<sup>†</sup>

<sup>\*</sup>Department of Electrical Engineering and Information Science,  
University of Science and Technology of China, Hefei, 230027, China

<sup>†</sup>Department of Computer Science, University of Texas at San Antonio, TX 78249, USA

## 5.12.1 Introduction

With the ever increasing popularity of digital devices equipped with cameras and the fast advance of Internet technology, billions of people are projected to the Web sharing and browsing photos. The ubiquitous access to both digital photos and Internet sheds bright light on many emerging applications based on visual search. Traditional visual search engines usually index multimedia visual data based on the surrounding textual information around images on the Web, such as titles and tags. Since textual information may be inconsistent with the visual content, content-based visual search is preferred and has been attracting lots of attention.

In content-based visual search, there exists the problem of *semantic gap*, which originates from the difficulty in describing high-level semantic concept with low-level visual feature. Instead of directly targeting at semantic-based visual search, most work in the multimedia community is focused on particular object/scene retrieval or partial-duplicate image retrieval. The former problem aims to retrieve all images containing a specific object or scene in a given query image from a large scale image database. The target images may undergo various changes and be taken under different views or imaging conditions. Typical ground truth dataset for this task includes the UKBench dataset [1], the Oxford Building dataset [2], and the Holidays dataset [3]. The goal of the latter problem is to find the partial-duplicate versions of the query with manual changes in color, scale, rotation, partial occlusion, compression rate, etc., in a large Web image database. The representative dataset for this task is the DupImage dataset [4] and the Copydays dataset [5].

In recent years, two pioneering works have paved the way to the significant advance in content-based visual retrieval on large-scale multimedia databases. The first one is the introduction of invariant local visual feature SIFT [6]. The SIFT feature is demonstrated with excellent descriptive and discriminative power in a variety of literature. It can well capture the invariance to rotation and scaling change and is robust to illumination change. The second work is the introduction of the Bag-of-Visual-Words (BoW) model [7]. Leveraged from information retrieval, the BoW model makes a compact representation of images based on the quantization of the contained local features and readily adapts to the classic inverted index structure for scalable image retrieval.

Based on the above pioneering works, the last decade has witnessed the emergence of numerous works on multimedia content-based visual search [1–4, 7–23]. Meanwhile, in industry, some commercial content-based search engines have been launched, such as Tineye [24] and Google Similar Image Search.

Technically speaking, there are generally three key issues in multimedia content-based retrieval: image representation, image organization, and image similarity formulation. Existing algorithms can also be categorized based on three key items.

**Image Representation:** The intrinsic problem in content-based visual retrieval is image comparison, before which an image is transformed to a kind of feature representation. The motivation is to achieve an implicit alignment so as to eliminate the impact of background and potential transformations or changes. In fact, how to represent an image is a fundamental problem in image understanding. There is a saying that “An image is worth more than a thousand words.” However, it is nontrivial to find those “words.” Usually, images are represented as a group of visual features. The representation is expected to be descriptive and discriminative. More importantly, it is also expected to be invariant to various transformations, such as rotation, resizing, and illumination change.

**Image Organization:** In multimedia retrieval, the visual database is usually very large. How to organize the large scale database to efficiently identify the relevant results of a given query is a nontrivial issue. Inspired by the success of information retrieval, many existing content-based visual retrieval algorithms and systems leverage the classic inverted file structure to index large scale visual database for scalable retrieval. To achieve this goal, visual codebook learning and feature quantization on high-dimensional visual features are involved.

**Image Similarity Formulation:** Ideally, the similarity between images should reflect the relevance in semantics, which, however, is difficult due to the “semantic gap” problem. Conventionally, the image similarity in content-based retrieval is formulated based on the visual feature matching results with some weighting schemes.

In the design of a multimedia content-based retrieval system, there are three key indicators which should be carefully considered: accuracy, efficiency, and memory cost. Usually, a retrieval method contributes to improving at least one of those indicators with little sacrifice in the other indicators.

**Accuracy:** In multimedia content-based visual retrieval, the retrieval quality (accuracy) of an image is measured by the well-accepted criterion of average precision, which takes consideration of both precision performance and recall performance. To obtain a high average precision, both precision performance and recall performance are required to be good enough. Any bias on precision or recall may lead to limited average precision and poor retrieval quality.

To evaluate the effectiveness of an algorithm, the mean average precision (mAP) defined in Eq. (12.1) is usually adopted, which takes the mean value of the average precision performance of multiple query images.

$$\text{mAP} = \frac{1}{Q} \sum_{i=1}^Q \frac{\sum_{k=1}^n P_i(k) \cdot \text{rel}_i(k)}{R_i}, \quad (12.1)$$

where  $Q$  denotes the number of query images,  $R_i$  denotes the number of relevant results for the  $i$ th query image,  $P_i(k)$  denotes the precision of top  $k$  retrieval results of the  $i$ th query image,  $\text{rel}_i(k)$  is an indicator function equaling 1 when the  $k$ th retrieval result of the  $i$ th query image is a relevant image and 0 otherwise, and  $n$  denotes the total number of returned results.

**Efficiency:** The efficiency of a retrieval system involves the time cost in visual vocabulary (or visual codebook) construction, visual feature indexing, and image querying. The first two items are performed off-line, while the last one is conducted on-line. Both the off-line and on-line processing is expected to be as fast as possible. Specially, the on-line querying is usually expected to be responded in real time.

**Memory Cost:** In a multimedia content-based visual retrieval system, the memory cost usually refers to the memory usage in the on-line query stage. Generally, the memory is mainly spent on the quantizer and the index file of database, which need to be loaded into the main memory for on-line retrieval. Popular quantizer includes tree-based structure, such as hierarchical vocabulary tree, randomized forests, etc., which usually cost a few hundred megabytes memory for codebook containing million-scale visual words. The index file size is proportional to the memory cost per indexed feature.

In the following section, we first briefly review the general pipeline of content-based visual search. After that, we discuss the five key modules of the pipeline, respectively. We also include some sample empirical results to illustrate the corresponding key points.

## 5.12.2 General pipeline overview

Content-based visual search or retrieval has been a core problem in multimedia for years. In recent literature, many approaches adopt invariant local features [6,25] to represent visual data, which exploit the Bag-of-Visual-Words (BoW) model [7] and the classic inverted index structure [26] for scalable image search. The general pipeline is illustrated in Figure 12.1. Such a visual search framework consists

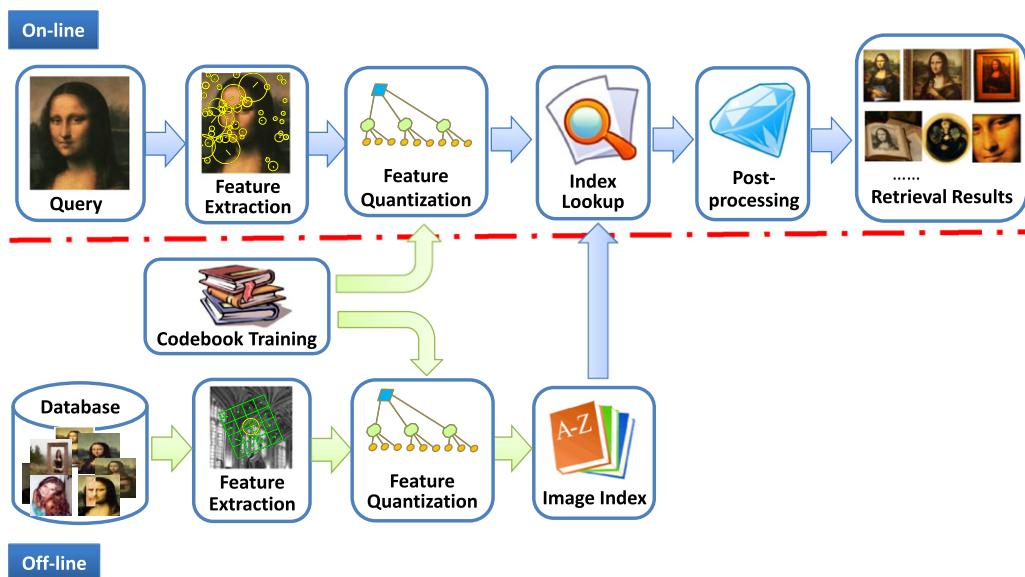


FIGURE 12.1

The general framework of BoW-based multimedia content-based visual retrieval.

of the off-line stage and the on-line stage, covering five necessary key modules, i.e., feature extraction, feature quantization, image indexing, retrieval scoring (in index lookup), and post-processing.

For feature extraction, the most popular and effective local descriptor is the SIFT [6], which is extracted on key points or regions detected by Difference of Gaussian (DoG) [6], MSER [27], or Hessian affine detector [25], etc. Later on, there have been lots of efforts in designing local descriptors with a higher efficiency and comparable discriminability, e.g., the SURF [28], KAZE [29], and Edge-SIFT [30]. At feature quantization, local visual features are mapped or hashed to a pretrained visual codebook and then an image is represented by a “bag” of visual words. Alternatively, hashing techniques, such as min-hashing [10] and geometric min-hashing [31], and packing method [32] can be applied to images’ visual word vectors to generate more compact representation. After this, inverted index structure is readily adopted to index large scale database for scalable visual search [7]. At the on-line retrieval stage, the shared visual words between a query image and database images can be easily identified by looking up the inverted index lists by quantizing the visual features in the query. The similarity between the query and the target database images is measured by a weighted formulation [1,33] based on those shared visual words. The initial retrieval results can be re-ranked by some post-processing techniques, such as the query expansion [8,13], feature augmentation [17], or geometric verification [2,12]. Finally, those relevant database images are ranked by their similarity scores and presented to users.

In the following sections, we make a review of related work in each module and discuss a variety of strategies to address the key issues in the corresponding modules.

### 5.12.3 Local feature representation

In content-based visual retrieval, images are usually represented by the extracted local visual features. Generally, local feature extraction involves two key steps, i.e., interest point detection and feature description. The detected interest points are expected to be highly repeatable over various transformations. After interest point detection, a descriptor is extracted to describe the visual appearance of the local patch centered at the interest point. Usually, the descriptor should be invariant to rotation and scale change, and robust to affine distortion, addition of noise, illumination changes, etc.

According to the data type and distance metric, local visual feature can be categorized into floating-point feature and binary feature. We discuss the representative features of each category in Sections 5.12.3.1 and 5.12.3.2, respectively. Generally, the floating-point local features are more descriptive and effective than the binary local features, although the binary features are characterized with the advantage of computing efficiency. To take advantage of both floating-point SIFT feature and binary feature, we introduce an efficient scheme to generate distance-preserving binary signature from SIFT, as discussed in Section 5.12.3.3.

#### 5.12.3.1 Floating-point local feature

The representative floating-point local feature is SIFT. For the standard SIFT feature [6] representation, key points are detected with the Difference-of-Gaussian (DoG) detector. A 128-D orientation histogram (SIFT descriptor) is extracted to capture the visual appearance of local patch centered at each key point. Before generating the final SIFT descriptor, the 128-D orientation histogram is unit-normalized and scaled by a constant factor. The DoG detector can also be replaced with MSER [27] or Hessian affine

[25]. As a variation of SIFT, SURF [28] is demonstrated with comparable performance but better efficiency in extraction.

The methods such as SIFT or SURF extract features in the Gaussian scale space with linear diffusion. However, the Gaussian blurring in SIFT does not respect the natural boundaries of objects and smoothes details and noise to the same degree. To address this issue, KAZE feature [29] operates completely in a nonlinear scale space. Through nonlinear diffusion, KAZE feature detects and describes features in nonlinear scale spaces keeping important image details and discarding noise when evolving the image in the scale space. It uses variable conductance diffusion. The nonlinear scale space is built efficiently via the stable and parallelizable Additive Operator Splitting (AOS) scheme. In the scenario of common 2D image matching applications [29], KAZE feature is demonstrated with notable improvement in repeatability and distinctiveness compared with SIFT and SURF.

### 5.12.3.2 Binary local feature

Recently, binary feature BRIEF [34] and its variants, such as ORB [35], FREAK [36], and BRISK [37], have been proposed and have attracted lots of attention in the computer vision community. The most notable merit of those binary features lies in the efficiency, which is over one order of magnitude faster than the classic SIFT feature. The feature detection of those binary features is based on the corner detector FAST [38], which is extremely efficient in implementation. The feature description of those binary features is based on sampling pixel pairs with some criteria from local patch for comparison to generate bit stream, which is also extremely efficient in implementation. Different from the above binary local features, Edge-SIFT [30] extracts binary local descriptors from the binary edge maps of scale- and orientation-normalized image patches. Since binary feature comparison can be efficiently performed with Hamming distance, binary features are promising to improve the efficiency in large scale retrieval. However, the dimensionality of those binary features is still very high, with 256-bit in ORB, 512-bit in both FREAK and BRISK, and 384-bit in Edge-SIFT. In the large search retrieval scenario with feature quantization involved, those binary features are revealed to be less effective than SIFT in the experimental study.

**Discussion:** In essence, the basic problem of image search is visual matching between images. When images are represented by local features, visual matching is achieved via feature matching between images. Intuitively, considering whether two features from different images are a valid match, the most straightforward criterion is to check whether the distance between them is smaller than a predefined threshold. In traditional Bag-of-Visual-Words based approach, feature matching is implicitly realized by checking whether two features are quantized to the same visual word. However, in large scale image search, many features with large distances from each other may be quantized to the same visual word, while many other features with small distance from each other are quantized to different visual words. Such phenomenon easily causes the false positive and true negative of local feature matches between images. To avoid such drawback, it is more preferable to verify feature matching by feature distance. Besides, since real-time response is a critical requirement in large scale image search, the matching verification should be performed very efficiently. To take advantage of the discriminative power of SIFT feature and the computing efficiency of binary feature, an alternative is to generate distance-preserving binary feature from SIFT feature. With this motivation, in the following section, we introduce a scheme on the generation of binary signature of SIFT [39].

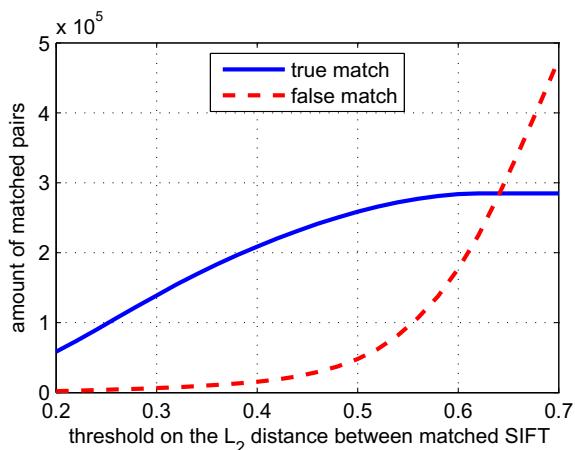
### 5.12.3.3 Binary signature of SIFT

Before introducing the Binary SIFT approach, in [Section 5.12.3.3.1](#), we show by experimental study that it is feasible to verify feature matching by the  $L_2$ -distance between SIFT features. In [Section 5.12.3.3.2](#), we introduce how to generate binary signature of SIFT. Experimental study reveals that Binary SIFT well keeps the feature distance of the original SIFT feature.

#### 5.12.3.3.1 $L_2$ -distance distribution of SIFT matches

In this section, we demonstrate that it is feasible to verify the SIFT matches between images by the threshold on the  $L_2$ -distances between features. To study the distribution of SIFT matches in terms of  $L_2$ -distance, we conduct an experimental study on about 5000 pairs of images, sampled from the DupImage dataset [4]. We test a series of different thresholds to select feature matches. If the  $L_2$ -distance between two unit-normalized features is less than the selected threshold, the two features are considered as a match. For each selected threshold, to identify those false matches, we adopt the geometric verification algorithm [4] to quickly select initial true matches as seeds to estimate an affine transformation to verify all matches. The frequency of true and false matches vs.  $L_2$ -distance thresholds is illustrated in [Figure 12.2](#).

From [Figure 12.2](#), we can observe that, when the distance threshold increases, the amount of identified true matches first steadily increases and then becomes stable. On the other hand, as the threshold value increases, the amount of identified false matches first keeps low and relatively stable, and then increases exponentially after the threshold becomes larger than 0.5. Such observation is due to the fact that the distance between true matches is generally smaller than that between false matches. When the threshold is low, few false matches are kept for geometric verification. And when the threshold grows, more true matches and more false matches pass the threshold for geometric verification. But when the threshold becomes too large, say over 0.55, few new true matches are included but the amount of false matches



**FIGURE 12.2**

The distribution of identified true matches and false matches based on  $L_2$ -distance thresholds.

grows exponentially. One conclusion drawn from Figure 12.2 is that a general threshold can be selected to distinguish the true matches and the false matches by making a tradeoff between including more true matches and excluding more false matches.

In visual matching based on SIFT descriptors, one effective criterion is to check the distance of the closest neighbor to that of the second-closest neighbor [6]. If the distance ratio is greater than a predefined threshold, the SIFT match is rejected as a false match. However, in large scale image search, given a local feature from a query image, it is infeasible to obtain the nearest and second nearest neighbors of local features in a target image. One feasible and reliable evidence that can be used is the  $L_2$ -distance from the nearest neighbors. If we can obtain an optimal threshold to distinguish true and false matches, we can exploit it to verify the  $L_2$ -distance between SIFT features. From Figure 12.2, the optimal threshold may be selected between 0.45 and 0.50. In fact, in large scale image search, it is even infeasible to directly compute the distance between the original SIFT features, since it is too memory-consuming to store the original SIFT features in the index file, let alone the high computational cost of  $L_2$ -distance computing. One solution to address this problem is to approximate the original SIFT feature with a new compact feature and verify the feature distance on the new features. The new feature shall be binary signature feature, so the feature distance can be efficiently computed and measured by Hamming distance. With such motivation, we propose the Binary SIFT, as discussed in detail in the next subsection.

#### 5.12.3.3.2 Binary SIFT generation

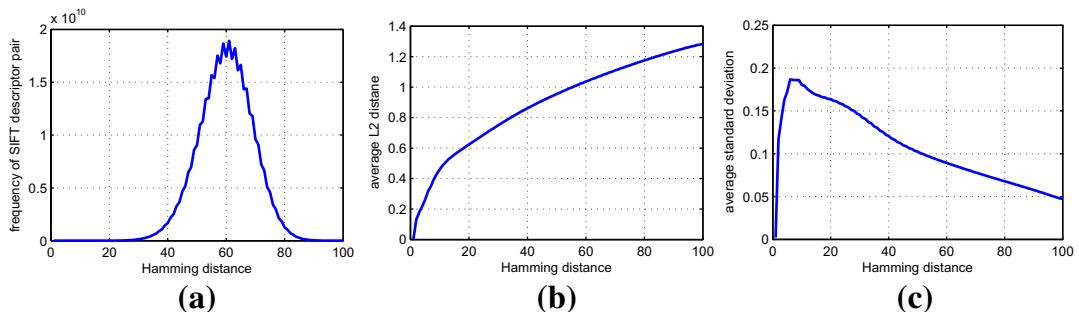
In SIFT descriptor ( $L_2$ -normalized 128-D vector [6]), each dimension corresponds to a bin of the concatenated orientation histograms. Generally, similar SIFT features have relatively smaller distances than irrelevant features. Features from the same source, e.g., image patch, may not be exactly the same due to image noise. But their values on the 128 bins usually share some common patterns, e.g., the pair-wise differences between most of bins are similar and stable. Therefore, it can be easily extended that the differences between bins and a predefined threshold are stable for most bins. Based on such observation, a scalar quantization strategy can be adopted to generate binary signature from SIFT.

Given a SIFT feature vector  $\mathbf{F} = (f_1, f_2, \dots, f_{128})^T$ , where  $f_i \in \mathcal{R}$ , a quantization function is defined to transform  $\mathbf{F}$  to a bit vector  $\mathbf{b} = (b_1, b_2, \dots, b_{128})^T$ , as follows:

$$b_i = \begin{cases} 1 & \text{if } f_i > \hat{f}, \\ 0 & \text{if } f_i \leq \hat{f}, \end{cases} \quad (1 \leq i \leq 128), \quad (12.2)$$

where  $\hat{f}$  is a threshold uniquely determined by the vector  $\mathbf{F}$ .

The threshold  $\hat{f}$  is an important parameter and determines the discriminative power of the generated binary signature. If the discriminative power of SIFT is well kept by Eq. (12.2), the Hamming distance between binary vectors  $\mathbf{b}$  should be consistent with the  $L_2$ -distance between original feature vectors  $\mathbf{F}$ . There may be many methods to choose the threshold  $\hat{f}$ , such as using the mean value of all bins in vector  $\mathbf{b}$  as  $\hat{f}$ , or learning  $\hat{f}$  from a training dataset. In the implementation,  $\hat{f}$  is selected as the median value of vector  $\mathbf{F}$ . The philosophy behind it is that, the median value is relatively stable to the coefficient change in some bins of a long vector. Another benefit as a byproduct from the median threshold selection is that, the obtained binary SIFT vector is implicitly normalized. With each high-dimensional feature quantized to a bit-stream vector, the feature comparison is transformed to the binary vector comparison, which can be efficiently accomplished by logical exclusive-OR operation and measured by Hamming distance.

**FIGURE 12.3**

The statistics on pairs of SIFT descriptors. (a) Descriptor pair frequency vs. Hamming distance; (b) the average  $L_2$ -distance vs. Hamming distance; (c) the average standard deviation vs. Hamming distance.

To demonstrate that the discriminative power of SIFT descriptors is well kept in the scalar quantization, a statistical study has been made on  $4.08 \times 10^{11}$  SIFT descriptor pairs, by including every SIFT pair extracted from image pairs randomly sampled from the UKBench dataset [1]. For each descriptor pair, its  $L_2$ -distance before scalar quantization and Hamming distance after scalar quantization are calculated. As shown in Figure 12.3a, the distribution of Hamming distances between these descriptors unsurprisingly exhibits a Gaussian-like distribution. From Figures 12.3b and c, it is observed that the Hamming distance between the quantized bit-vectors is consistent with the average  $L_2$ -distance, with relatively small standard deviation (computed on the unit-normalized descriptors). To further reduce the deviation, we use a variation of Eq. (12.2) and transform the descriptor vector to a 256-bit vector, which will be discussed at the end of this section.

It should be noted that the above binary SIFT generation approach is different from the SIFT quantization methods proposed in lattice quantization [40] and Hamming Embedding [3]. In [40], the descriptor space is arbitrarily split along dimension axes into regular lattices. In [3], for each bin/dimension, a median value of all training features on that bin in the low-dimensional space is computed for binarizing the corresponding dimension. Both two approaches ignore the unique property of every individual SIFT descriptor.

Figure 12.4 shows a real instance of local descriptor match across two images based on binary SIFT. From Figure 12.4b, it can be observed that these two SIFT descriptors have similar magnitude in the corresponding bins with some small variations before quantization. After the scalar quantization, they differ from each other in nine bins. With a proper threshold, it can be easily determined whether the local match is true or false just by the exclusive-OR (XOR) operation between the quantized bit-vectors. Obviously, the error in the exclusive-OR result is likely to occur in those bins with magnitude around the median value. Intuitively, the median threshold could be increased to some upper level, which can make the Hamming distance between similar SIFT descriptors smaller. However, such modification will also reduce the Hamming distance between irrelevant descriptors and weaken the discriminative power of the binary SIFT.

A statistical study on the distribution of median value of SIFT descriptor has been performed. One hundred million SIFT descriptors are sampled from a large dataset, and the median value of each 128-D

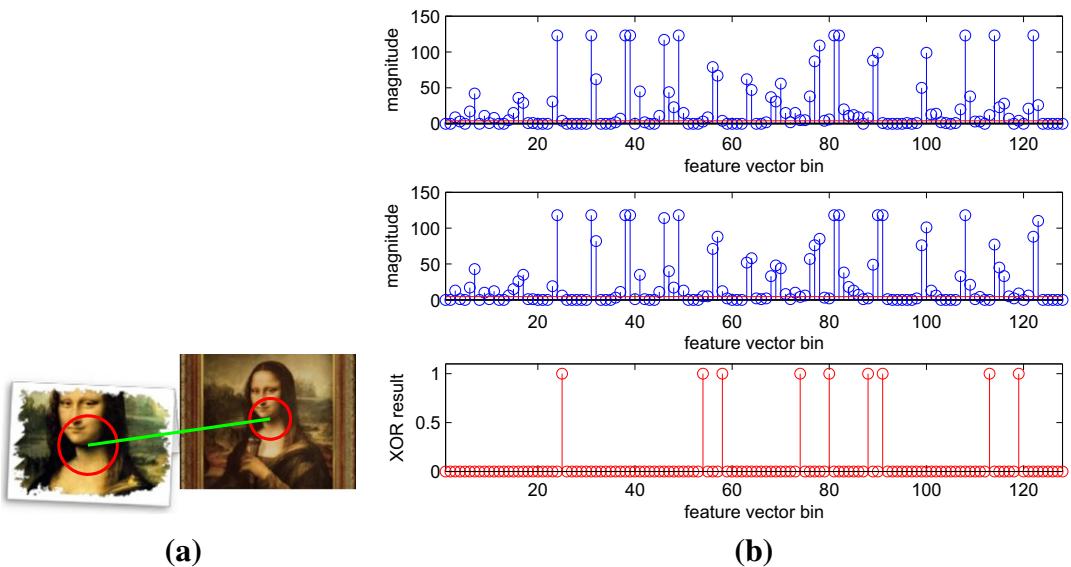


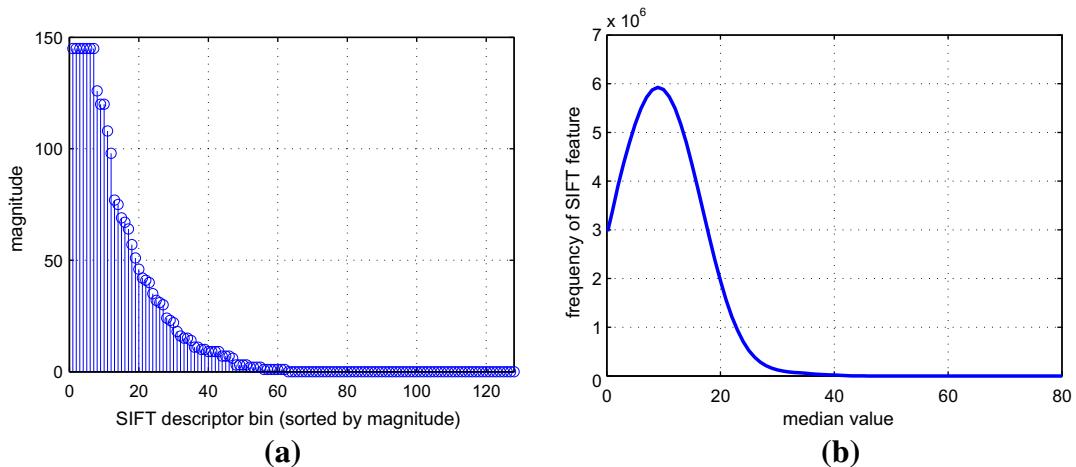
FIGURE 12.4

Example of feature matches. (a) A local match between two images. The endpoints of the green line denote the key point positions of two SIFT features. The radius of the red circle centered at the key points is proportional to the SIFT features characteristic scale. (b) Top: the 128-D descriptor of the matched SIFT feature in the left image; middle: the 128-D descriptor of the matched SIFT feature in the right image; bottom: the XOR result of the binary SIFT features from the two matched SIFT features. The red horizontal lines in the top and bottom figure denote the median values of the two SIFT descriptors, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this book.)

descriptor vector is computed. As shown in Figure 12.5, the median value of most SIFT descriptors is relatively small, around 9, but the maximum magnitude in some bins still can reach more than 140. This may incur potential quantization loss since those bins with magnitude above the median are not well distinguished. To address this issue, the same scalar quantization strategy could be conducted again on those bins with magnitude above the median. Intuitively, such operation can be performed recursively. However, it will cause additional storage cost. In the implementation, the scalar quantization is performed twice, i.e., first on the whole 128 elements, and then on those elements with magnitude above the median value. Consequently, a SIFT descriptor  $\mathbf{F} = (f_1, f_2, \dots, f_{128})^T$  is quantized to a 256-bit vector  $\mathbf{b} = (b_1, b_2, \dots, b_{256})^T$ , as follows:

$$(b_i, b_{i+128}) = \begin{cases} (1, 1) & \text{if } f_i > \hat{f}_2, \\ (1, 0) & \text{if } \hat{f}_1 < f_i \leq \hat{f}_2, \\ (0, 0) & \text{if } f_i \leq \hat{f}_1, \end{cases} \quad (12.3)$$

where  $\hat{f}_1 = \frac{g_{64} + g_{65}}{2}$ ,  $\hat{f}_2 = \frac{g_{32} + g_{33}}{2}$ ,  $(g_1, g_2, \dots, g_{128})$  is the sorted vector from  $(f_1, f_2, \dots, f_{128})$  in the descending order. With Eq. (12.3), each dimension of SIFT descriptor is divided into three parts,

**FIGURE 12.5**

Statistics of SIFT descriptors. (a) A typical SIFT descriptor with bins sorted by magnitude in each dimension; (b) the frequency distribution of median value of the descriptor vector among 100 million SIFT descriptors.

**FIGURE 12.6**

Examples of local matching results based on 256-bit BSIFT. The hamming distance threshold is selected as 24. No other geometric verification is involved. The lines endpoints denote the key point positions of two SIFT features. The radius of the red circle centered at the key point is proportional to the SIFT features characteristic scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this book.)

and 2 bits are used to encode each part. It should be noted that there is still some redundancy with such representation. According to the Shannon's information theory, the bit rate of binary SIFT by Eq. (12.3) is 1.5 bits per bin. Therefore, it may only take 196 bits to represent each BSIFT if compression is required.

Some sample results of image local matching based on BSIFT by Eq. (12.3) are illustrated in Figure 12.6, from which we can observe that those true local matches are satisfactorily identified even without introducing any false match.

With the transformation by Eq. (12.3), the comparison of SIFT descriptors by  $L_2$ -distance is captured by the Hamming distance of the corresponding 256-bit vectors. Since the target is large scale image

search, how to adapt the scalar quantization result to the classic inverted file structure for scalable image search needs to be explored. We discuss this issue in [Section 5.12.5.2](#).

The potential concern of the binary SIFT is the high dimensionality. To partially address this issue, PCA can be introduced to reduce the dimension of the original SIFT descriptor before performing the scalar quantization. After the dimension reduction, binary signature can be generated based on the low-dimensional feature, resulting in low bit-rate binary signature.

## 5.12.4 Feature quantization

In multimedia content-based retrieval, local visual features are often high-dimensional, and hundreds or thousands of local features can be extracted from a single image. To achieve a compact representation of each image and flexibly adapt to the inverted index structure leveraged from information retrieval, it is necessary to perform feature quantization. Analogous to the text codebook in information retrieval, a visual codebook is usually constructed off-line. Then, feature quantization maps a high-dimensional feature to the visual codebook. The most popular feature quantization method is vector quantization. As an alternative, hashing approach can also be used. In the following, we review the existing techniques on vector quantization in [Section 5.12.4.1](#) and introduce a cascaded hashing method for feature quantization in [Section 5.12.4.2](#).

### 5.12.4.1 Vector quantization

Vector quantization models the probability density functions by the distribution of high-dimensional visual feature vectors. To explore the feature distribution, sufficient training feature samples are collected and clustering techniques, such as  $k$ -means, hierarchical  $k$ -means, and approximate  $k$ -means (AKM) [2], can be adopted to identify the density of high-dimensional visual data. As a result, a group of cluster centers are generated and regarded as visual words, constituting a visual codebook.

With visual codebook defined, feature quantization assigns each visual feature with the ID of the closest visual word in the high-dimensional feature space. The most naive choice is to find the closest (the most similar) visual word of a given feature by linear scan, which, however, suffers expensive computational overhead. Usually, approximate nearest neighbor (ANN) search methods are adopted to speed up the searching process, with sacrifice of accuracy to some extent. In [6], a  $k$ -d tree structure [41] is utilized with a best-bin-first modification to find approximate nearest neighbors to the descriptor vector of the query. In [1], based on the hierarchical vocabulary tree, an efficient approximate nearest neighbor search is achieved by propagating the query feature vector from the root node down the tree by comparing the corresponding child nodes and choosing the closest one. In [42], a  $k$ -d forest approximation algorithm is proposed with reduced time complexity. To reduce the quantization loss, a descriptor-dependent soft assignment scheme [9] is proposed to map a feature vector to a weighted combination of multiple visual words. In [40], the high-dimensional SIFT descriptor space is partitioned into regular lattices. Although demonstrated to work well in image classification, in [9], regular lattice quantization is revealed to work much worse than [1, 9] in large scale image search application.

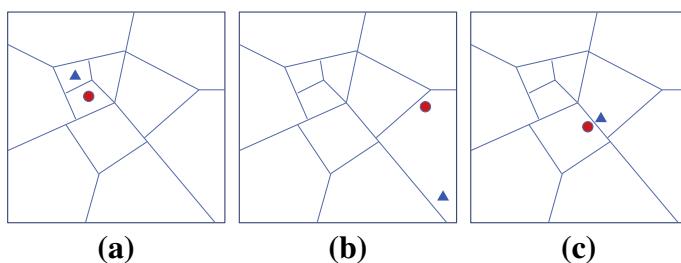
In [3], Hamming Embedding reduces the dimension of SIFT descriptors quantized to a visual word, and then trains a median vector by taking the median value in each dimension of the feature samples. After a new feature is quantized to a visual word, it is projected to the low-dimensional space, and then

compared with the median vector dimension-wise to generate binary signature for matching verification. In [43], its variation, i.e., the asymmetric Hamming Embedding scheme, is proposed to exploit the rich information conveyed by the binary signature.

The vector quantization of local descriptors is closely related to approximate nearest neighbor search. In the literature, there are many hashing algorithms for approximate nearest neighbor search, e.g., LSH [44], kernelized locality sensitive hashing [45], semi-supervised hashing method (SSH) [46], spectral hashing [47], min-Hashing [10], iterative quantization [48]. These hashing methods, however, are mostly applied to global image features such as GIST or BoW features at the image level, or to feature retrieval only at the local feature level. There are few works on image level search based on local feature hashing [16]. This is mainly due to the fact that those hashing schemes are capable of achieving a high precision but without guarantee on the recall rate, which may not benefit the final retrieval accuracy. To obtain a relatively high recall rate of feature matching, those hashing schemes have to generate tens of or hundreds of hashing tables, which will require a heavy memory cost for each indexed local feature of database images and meanwhile consume much more time during retrieval.

#### 5.12.4.1.1 Discussion on vector quantization

Although great success has been witnessed with vector quantization in large scale content-base visual retrieval, there are two issues led by a large visual codebook. First, the visual codebook requires considerable resources to train off-line and consumes large amount of memory on-line. For example, a hierarchical codebook with a million visual words for 128D SIFT descriptors [6] is generally learned from tens of million training descriptors, and requires several hundred megabytes to store at the runtime. These requirements prohibit its usage in resource limited scenarios, e.g., indexing and searching 10,000 images locally on a mobile device. Second, the feature quantization error using a hierarchical vector quantization is not easy to control. Depending on the training descriptors, a large codebook divides the feature space to multiple small cells (i.e., the hashing cells) with a variable coverage. It is not rare to observe such cases shown in [Figure 12.7](#): features at a small distance from each other are quantized to



**FIGURE 12.7**

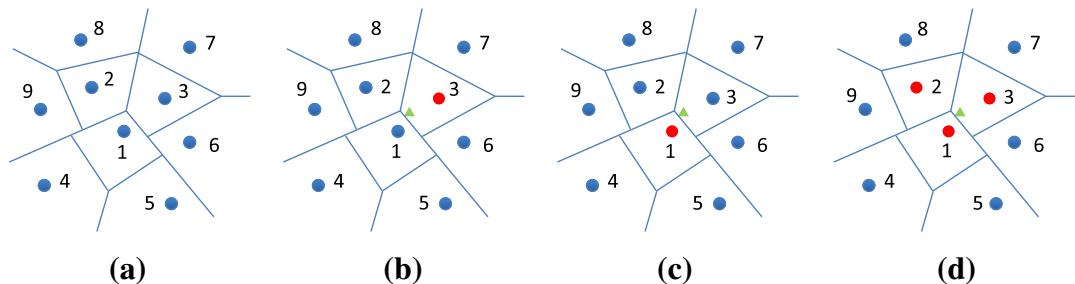
Mock-up illustration of the issues in the codebook-based vector quantization. The feature space is split into multiple small cells, each of which corresponds to a visual word. The query feature and candidate database feature are denoted by a red circle and a green triangle, respectively. (a) Two features at a small distance are quantized to different visual words; (b) two features at a large distance are quantized to the same visual word; (c) two features close to the cell boundary are quantized to different visual words. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this book.)

different words in Figure 12.7a; features at a large distance from each other are quantized to the same word in Figure 12.7b; or features close to quantization cell boundaries are separated to different words due to the hard-decision strategy, no matter how small their distances are, as illustrated in Figure 12.7c. These cases easily cause the false positive or false negative matches of local features, leading to less precise retrieval results.

#### 5.12.4.1.2 Visual word expansion

In the BoW model, feature quantization is usually performed in a hard-decision mode or with a soft assignment strategy. In the hard-decision quantization based on visual vocabulary tree [1], given a new feature, we traverse from the root node and go down along the nearest child node, until reaching a leaf node. However, such leaf node does not necessarily correspond to the nearest visual word, especially when the test feature is near the cell boundary of visual words, as the instance shows in Figure 12.8b. In fact, some approximate-nearest-neighbor search algorithms, such as  $k$ -d tree, can be used to find a better visual word, which is more likely to be the nearest visual word of the codebook to the test feature. As a result, the quantization error can be reduced and better retrieval quality can be expected. However, such approaches are usually computationally expensive. To address this dilemma, we introduce a visual word expansion scheme [49], which reduces quantization loss but introduces little computational overhead.

The visual word expansion scheme is based on the observation that, given a test feature vector, the expected nearest visual word is always close to the approximate visual words obtained by ANN search, such as hierarchical  $k$ -NN search [1]. Therefore, we can build a table to record the nearest visual words of each visual word in the visual vocabulary beforehand. That is, for each visual word, we find the top  $p$  nearest visual words, called *supporting visual words*, in the codebook by  $k$ -d tree [50]. Each visual word itself is also considered as its supporting visual word. Such processing can be efficiently performed off-line.



**FIGURE 12.8**

A toy example of visual word expansion with  $p = 4$  and  $k = 3$ . (a) Nine visual words are obtained by hierarchically clustering training feature samples. Each blue circle denotes a visual word; (b) a new test feature (green triangle) is quantized to visual word 3 by hierarchical  $k$ -NN search; (c) visual word 1 is found as the nearest visual word by checking the 4 nearest neighbors of visual word 3; (d) visual word 1, 2, and 3 are identified as the nearest visual words to the test feature by checking the 4 nearest neighbors of visual word 1. (Best viewed in color PDF).

Given a test feature  $\mathbf{f}^{(i)}$ , after finding the approximate visual word  $\mathbf{v}_i$  by traversing the hierarchical vocabulary tree [3], we further compare the test feature vector with the  $p$  supporting visual words  $\{\mathbf{v}_{i,j}, j = 1, \dots, p\}$  of the  $\mathbf{v}_i$ . Then, the  $p$  supporting visual words are sorted by their distances from the test feature vector in ascending order. When the test feature is an indexed feature of a database image, we just take the nearest supporting visual word  $\mathbf{v}_{i,k}$  as the quantization result of the test feature, which satisfies:

$$\|\mathbf{v}_{i,k} - \mathbf{f}^{(i)}\| < \|\mathbf{v}_{i,j} - \mathbf{f}^{(i)}\|, \quad \forall j : 1 \leq j \leq p, \quad j \neq k. \quad (12.4)$$

On the other hand, when the test feature is from a query image in the on-line search stage, we continually compare the test feature  $\mathbf{f}^{(i)}$  with those supporting visual words of visual word  $\mathbf{v}_{i,k}$ . Then, we select the top  $k$  supporting visual words of  $\mathbf{v}_{i,k}$  as the quantization results of  $\mathbf{f}^{(i)}$  and regard those indexed features in the inverted image list following those supporting visual words as candidate matches to the query feature  $\mathbf{f}^{(i)}$ . Such processing will increase the retrieval recall performance.

Intuitively, we can also adopt the soft assignment strategy to index each database feature with multiple nearest supporting visual words. However, such processing will increase multiple times memory cost per indexed feature. Therefore, in the implementation, we can only index each feature into the inverted feature list of the nearest supporting visual words.

[Figure 12.8](#) illustrates a toy example of the visual word expansion idea for quantization. [Figure 12.8a](#) shows a tree, which is hierarchically built with the branch number as 3 and the depth as 2, and the 2D feature space is represented by 9 visual words. In [Figure 12.8b](#), given a test feature (green triangle), we quantize it to visual word 3 by traversing the vocabulary tree. Then, by comparing the distance between the test feature and the 4 supporting visual words of visual word 3, we can easily identify that the nearest visual word is visual word 1 as shown in [Figure 12.8c](#), which is used to index the test feature. When the test feature is a query feature of a query image, we will check the inverted image lists of the top 3 nearest supporting visual words (visual word 1, 2, and 3) of visual word 1, as highlighted in red in [Figure 12.8d](#).

### 5.12.4.2 Scalable cascaded hashing

To avoid the above issues in the codebook-based vector quantization, an alternative is scalable cascaded hashing (SCH), which hashes high-dimensional visual features without involving any codebook training and vector quantization. We first ensure the recall rate of feature matching with a cascaded hashing scheme which conducts scalar quantization on the principal components of local descriptors in a cascaded manner. Then, we improve the precision rate of feature matching by verifying the binary signatures of local descriptors, which effectively bounds the quantization error to the same hashing value. The recall and precision achieved by the above two steps will eventually boost the retrieval accuracy.

Scalable cascaded hashing follows the strategy of first ensuring a relatively high recall rate of local feature matching and then refining the matching to improve the precision rate. We first conduct a PCA for dimension reduction on SIFT features [6] in [Section 5.12.4.2.1](#). We ensure the recall rate of local feature matching by cascaded hashing of the principal components of SIFTs as discussed in [Section 5.12.4.2.2](#). Ensuring the recall rate inevitably incurs some false positive feature matches. To address this issue, in [Section 5.12.4.2.3](#), the candidate feature matches can be verified by compact binary signatures of SIFT descriptors, which effectively remove a large portion of false positive matches and greatly improves the precision rate. Consequently, promising retrieval accuracy can be achieved based on reliable feature matching, free of any codebook.

#### 5.12.4.2.1 Dimension reduction on SIFT by a PCA

Before conducting the cascaded hashing, the dimension of SIFT feature is reduced by a PCA. The major benefit from a PCA is that the top  $k$  principal dimensions of PCA preserve maximum energy of the original descriptor. Therefore, we can only focus on those top few dimensions instead of all 128 dimensions to reduce processing complexity. These low-dimensional features facilitate a high recall rate with a limited number of hashing operations. We have collected 5 million SIFT features for the PCA training. Those training features are randomly sampled from a 50-million feature set, which are extracted from an independent image database. We denote the dimension-reduced SIFT feature as PSIFT, to distinguish from the PCA-SIFT [51]. Some interesting observations are revealed from the results shown in Figure 12.9.

As illustrated in Figure 12.9a and b, the energy (eigenvalue of PCA) of SIFT feature is concentrated on a few dimensions. Figure 12.9d shows the coefficient distributions for the top 20 dimensions. It is worth noting that the coefficient distribution for the top 1 dimension exhibits a mixture of two Gaussian-like distributions, while for other dimensions the coefficient distribution presents a single Gaussian-like distribution. Besides, the coefficient range in each dimension is also different from each other. Some existing works, such as [3, 15, 48], assign 1 bit independently to quantize each dimension of the transformed SIFTS. However, as demonstrated in Figure 12.9c and Figure 12.9d, 1 bit may be far from enough to encode a dimension if the hashing is conducted independently for each dimension. For instance, the entropy of dimension 2 is about 8.7, which means at least 8.7 bits are required to encode the distribution without too much quantization error. Even if some error is allowed, it is still too rough to independently describe such a single Gaussian-like distribution with only 1 bit. In other words, these statistics suggest us to assign multiple bits to each dimension, which means multiple quantization steps for one dimension in case of a scalar quantization, unless the context among different dimensions is explored as in [39]. Based on the above observations, in the following section, a cascaded hashing scheme is introduced splitting the value ranges of PSIFT's top dimensions, which largely assures the feature matching at certain given recall rates.

#### 5.12.4.2.2 Cascaded hashing

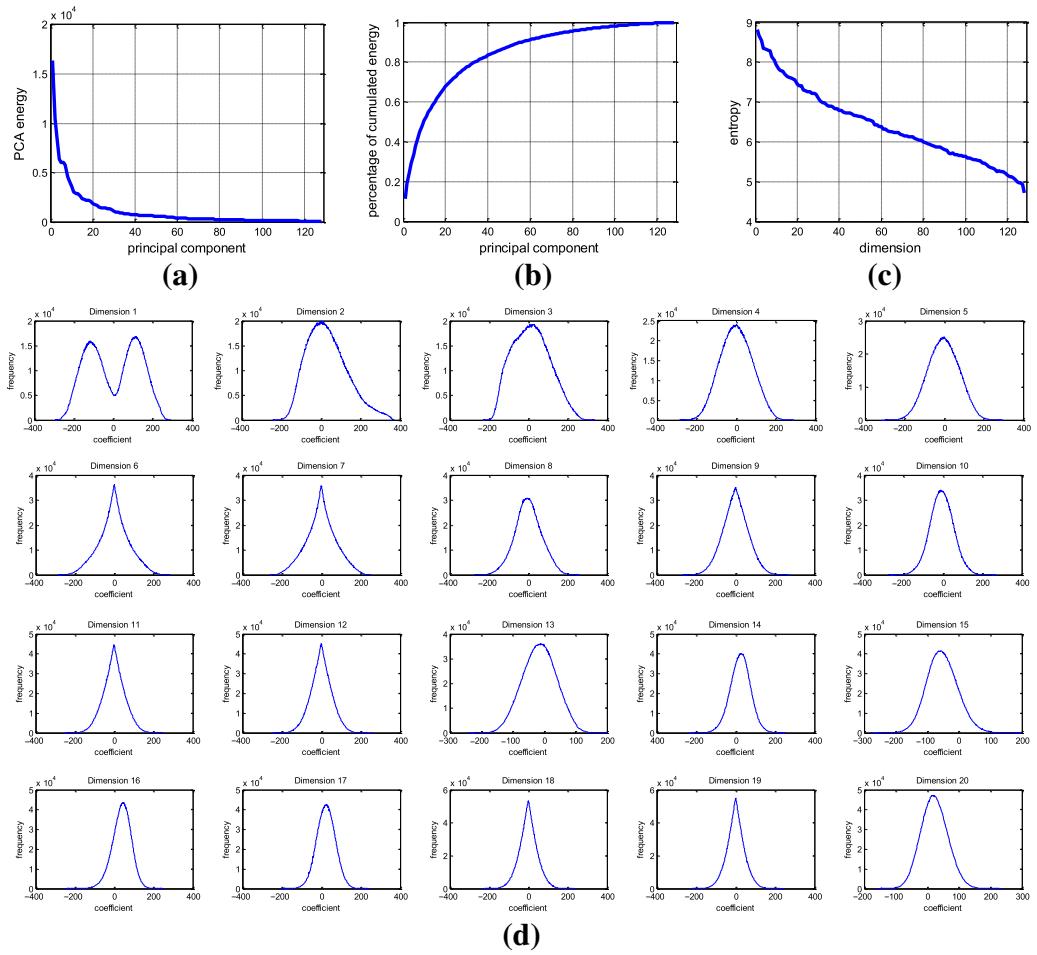
The general strategy of the scalable cascaded hashing approach (SCH) is to first ensure recall rate and then improve the precision rate in feature matching. In the cascaded hashing, scalar quantization is sequentially performed on the principal components of SIFT such that the accumulative recall rate is relatively high while the false positive rate is low. The SCH scheme can be regarded as an approximate nearest neighbor search method focusing on ensuring the recall rate of local feature matching. Denote a PSIFT data point as  $\mathbf{y} \in \mathcal{R}^d$  and a PSIFT query as  $\mathbf{q} \in \mathcal{R}^d$ , then  $\mathbf{q}$ 's  $\epsilon$ -neighborhood is given as,

$$NN(\mathbf{q}, \epsilon) = \{\mathbf{y} \mid \|\mathbf{q} - \mathbf{y}\|_2 < \epsilon\}. \quad (12.5)$$

Since the PCA projection is orthogonal and preserves the  $L_2$ -distance of original SIFT descriptors [15], Eq. (12.5) is an approximation of the  $\epsilon$ -neighborhood of the corresponding feature in the original SIFT space.

Denote  $\mathbf{y}^k$  as the vector of the top  $k$  dimensions of  $\mathbf{y}$  and define  $\mathbf{q}^k$  for  $\mathbf{q}$  in the same way. We relax Eq. (12.5) in the following way denoting the approximate nearest neighbor set,

$$AN(\mathbf{q}^k, \mathbf{t}) = \{\mathbf{y}^k \mid \|q_i - y_i\|_2 < t_i, i = 1, 2, \dots, k\}, \quad (12.6)$$

**FIGURE 12.9**

PCA results on 5 million SIFT training samples. (a) The energy (eigenvalues of PCA) corresponding to each principal component; (b) the cumulated energy distribution over principal components; (c) the entropy of each dimension on the coefficient distribution after the dimension reduction; (d) the coefficient distributions of the top 20 dimensions in PSIFT.

where a series of thresholds  $\mathbf{t} = \{t_i\}$  on each dimension are critical in the cascaded hashing scheme. We determine the thresholds by extensive empirical study with the constraints on the recall rate of local feature matching. The threshold  $t_i$  for the  $i$ th dimension is sequentially determined with the expected recall rate of the candidate results:

$$t_i = \operatorname{argmin}_n \int_0^n p_i(x) dx > r_i, \quad (12.7)$$

where  $p_i(x)$  denotes the probability density function of absolute coefficient distance between relevant features (truly matched features under some criteria) for the  $i$ th dimension,  $r_i$  is the relative recall rate for the  $i$ th dimension defined as,

$$r_i = \begin{cases} \frac{|NN(\mathbf{q}, \epsilon) \cap AN(\mathbf{q}^i, \mathbf{t})|}{|NN(\mathbf{q}, \epsilon)|}, & \text{for } i = 1, \\ \frac{|NN(\mathbf{q}, \epsilon) \cap AN(\mathbf{q}^i, \mathbf{t})|}{|NN(\mathbf{q}, \epsilon) \cap AN(\mathbf{q}^{i-1}, \mathbf{t})|}, & \text{for } i > 1. \end{cases} \quad (12.8)$$

The relative false positive rate in the  $i$ th dimension is defined as,

$$f_i = \begin{cases} \frac{|AN(\mathbf{q}^i, \mathbf{t}) \setminus NN(\mathbf{q}, \epsilon)|}{|\mathcal{S}|}, & \text{for } i = 1, \\ \frac{|AN(\mathbf{q}^i, \mathbf{t}) \setminus NN(\mathbf{q}, \epsilon)|}{|AN(\mathbf{q}^{i-1}, \mathbf{t}) \setminus NN(\mathbf{q}, \epsilon)|}, & \text{for } i > 1, \end{cases} \quad (12.9)$$

where  $\mathcal{S}$  denotes the set of all potential feature matches. In the SCH, we do not explicitly constrain the false positive rate but focus on the recall rate.

So the overall recall after cascaded quantizing  $c$  dimensions ( $c < k$ ) is expressed as

$$\text{recall}(c) = \prod_{i=1}^c r_i = \frac{|NN(\mathbf{q}, \epsilon) \cap AN(\mathbf{q}^c, \mathbf{t})|}{|NN(\mathbf{q}, \epsilon)|}. \quad (12.10)$$

Thus the overall false positive rate after quantizing  $c$  dimensions ( $c < k$ ) is

$$FP(c) = \prod_{i=1}^c f_i = \frac{|AN(\mathbf{q}^c, \mathbf{t}) \setminus NN(\mathbf{q}, \epsilon)|}{|\mathcal{S}|}. \quad (12.11)$$

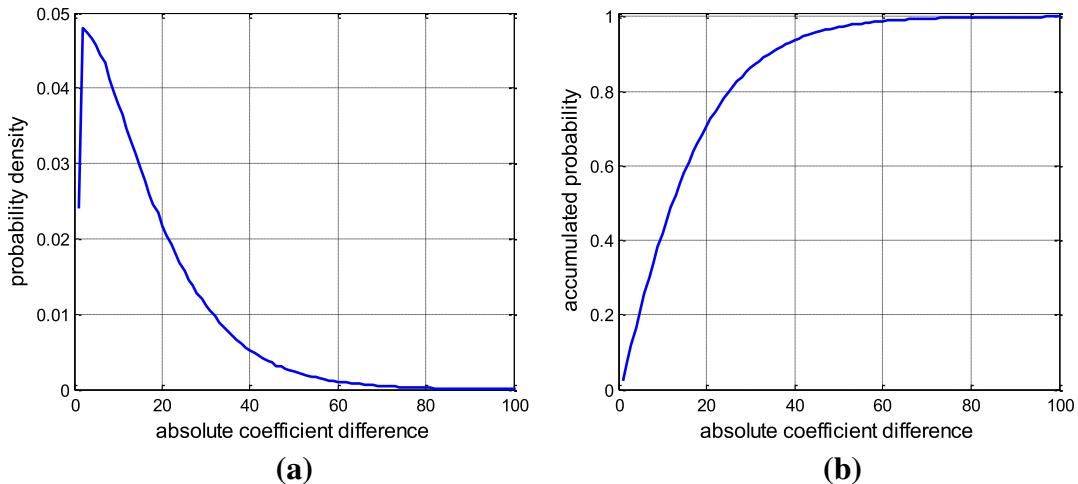
To ensure the overall recall in Eq. (12.10) large enough, we impose the constraint on the recall rate of each of the  $c$  dimensions:

$$r_i > \xi. \quad (12.12)$$

Therefore, we have  $\text{recall}(c) > \xi^c$ . The selection of  $\xi$  impacts both the overall recall rate  $\text{recall}(c)$  and the overall false positive rate  $FP(c)$ . For instance, if we select  $\xi = 0.95$  (which means 95% true relevant matches are kept from the previous round of scalar quantization) and  $c = 10$ , we have  $\text{recall}(c) > \xi^c \approx 0.60$ . Meanwhile, the selection of  $\xi$  should remove the vast majority of irrelevant false matches after the sequential filtering. This strategy shares some insights with the face detection algorithm [52], which filters irrelevant feature samples gradually in a cascaded manner.

To select the threshold  $t_i$  in Eq. (12.6) with the predefined constraint on the recall rate, we need to explicitly identify the probability density function  $p_i(x)$  in Eq. (12.7). To achieve the goal, we pair-wisely select two images from relevant image groups to generate about 15K image pairs, leading to 1.54 million relevant feature pairs. Based on those feature pairs, we conduct the feature transformation by a PCA and build the probability density function  $p_i(x)$  on the absolute coefficient distance between relevant features for each dimension of PSIFT.

Figure 12.10a illustrates the probability density function for the first dimension. The accumulated probability over  $p_i(x)$  is shown in Figure 12.10b. From Figure 12.10b, we observe that about 94%

**FIGURE 12.10**

Distribution of the absolute coefficient difference for the top 1 dimension of the PSIFT on the 1.54 million pairs of relevant feature pairs. (a) Probability density distribution of the absolute coefficient difference; (b) the accumulated probability integrated based on (a).

relevant true matches are kept if we set the difference threshold as 40. Considering the large range of the coefficient in the corresponding dimension, the portion of false matches within the threshold is relatively small. Based on such observations, if we sequentially cascade  $k$  dimensions and select a proper threshold at each dimension to keep a high recall from the previous round of scalar quantization, both the overall recall and false positive rate decrease exponentially. Importantly, the false positive rate decreases much faster, which benefits narrowing down the search scope of candidate feature samples.

In the following, we present the cascaded hashing scheme for the indexed features and query feature, respectively. Given a feature vector  $\mathbf{q}^c$ , which is the sub-vector of the top  $c$ -dimension from a PSIFT, its cascaded hashing result, referred to as the SCH vector, is defined as the concatenation of scalar quantization result at each dimension, i.e.,

$$\text{SCH}(\mathbf{q}^c) = (h_1, \dots, h_c), \quad (12.13)$$

where the quantization result in the  $i$ th dimension is defined as,

$$h_i = \lfloor \frac{q_i - m_i}{s_i} \rfloor, \quad 1 \leq i \leq c, \quad (12.14)$$

where  $m_i$  denotes the minimum coefficient in the  $i$ th dimension, and  $s_i$  denotes the quantization step in the  $i$ th dimension.

If the feature vector is from a database image, a hash key is generated from the cascaded quantization result in Eq. (12.13) for indexing based on the inverted index structure. If the feature vector is from a query image, we will take a different strategy to reduce the quantization error as discussed below.

Based on Eq. (12.14), each dimension is uniformly split into multiple cells. Denote the center of the  $j$ th cell in the  $i$ th dimension is  $h_i^{(j)}$ , which is a scalar. To tolerate the quantization error from the hard-decision strategy in quantization by Eq. (12.14), we define the following criterion to identify the candidate feature matches of a query feature. Given a PSIFT query feature  $\mathbf{q}^c \in \mathcal{R}^c$ , we perform quantization on each dimension  $q_i$  of  $\mathbf{q}^c$  with a soft-decision strategy, by selecting those cells close to  $q_i$  as follows:

$$|q_i - h_i^{(j)}| < s_i, \quad \forall j, \quad 1 \leq j \leq c. \quad (12.15)$$

To ensure the recall performance in Eq. (12.12), in the implementation, we define  $s_i = 2 \cdot t_i$ , because, in image search, all features of database images have to be quantized to a hash key for the convenience of index and scalable retrieval. After the feature indexing, the original features of the database images are discarded. Therefore, for each indexed feature, we only know the cell it is located in the  $i$ th dimension by the scalar quantization result in Eq. (12.14), but do not know the precise location in the corresponding cell. Give a query feature, we relate the  $i$ th dimension with two closest cells. To make sure that the two closest cells contain all relevant indexed features with the absolute difference in the  $i$ th dimension less than  $t_i$ , the best choice is to set  $s_i$  to be  $2 \cdot t_i$ .

Using Eq. (12.15), each dimension of the query feature  $\mathbf{q}^c$  is assigned the ID of at most 2 cells. That is, there are at most two alternative quantization results in each dimension for the query feature. Then the final SCH result of  $\mathbf{q}^c$  is obtained by alternatively selecting one quantization cell in each dimension. So each query feature is quantized to at most  $2^c$  SCH vectors. For a query feature, all features indexed to any of these SCH vectors are considered as candidate matches.

The retrieval efficiency of the cascaded hashing approach can be improved if the number of hash buckets is reduced. In the above discussion, the quantization is performed in each dimension independently. Therefore, with  $c$  dimensions, the maximum number of nonempty hash buckets is  $2^c$ . However, if in some dimension the query coefficient is located close to the cell center, we may only check one cell instead of two, with minor sacrifice of accuracy. If  $m$  out of  $c$  dimensions in a query feature satisfy the above condition, it is possible to reduce the bucket number from  $2^c$  to  $2^{(c-m)}$ , which causes some minor reduction in accuracy but may greatly improve the efficiency.

### 5.12.4.2.3 Matching verification by binary signatures

In Section 5.12.4.2.2, we have discussed the cascaded hashing scheme which guarantees the rate of true positive results but inevitably leads to some false positive matches. To identify and remove those false positive results, it is necessary to perform matching verification on these candidates. To make the verification fast enough, it is preferable to transform the feature to binary code and verify the matching with the efficient Hamming distance measurement. Motivated by this, we adopt the binary signature generation approach as discussed in Section 5.12.3.3.2 for those dimensions in PSIFT after the top  $c$  dimensions. In other words, for the PSIFT as  $\mathbf{y} \in \mathcal{R}^d$ , we select a sub-set of elements in  $\mathbf{y}$  and obtain the vector  $\mathbf{z} \in \mathcal{R}^e$ , with  $z_i = y_{c+i}$ , considering that the top  $c$  dimensions have already been used in the scalable cascaded hashing in Section 5.12.4.2.2. Then, we transform the vector  $\mathbf{z}$  to a binary vector  $\mathbf{b} = (b_1, \dots, b_e)$  by comparing each coefficient with an individual threshold as follows,

$$b_i = \begin{cases} 1 & \text{if } z_i > \hat{z}, \\ 0 & \text{if } z_i \leq \hat{z}, \end{cases} \quad (12.16)$$

where  $\hat{z}$  is the median of all dimensions for an individual vector  $\mathbf{z}$ , as discussed in [Section 5.12.3.3.2](#). Different feature vectors will have different median values, and the median value  $\hat{z}$  of each feature  $\mathbf{z}$  is computed on-line. The rationale behind [Eq. \(12.16\)](#) is that the relative coefficient differences between different dimensions are assumed to be stable. Unlike [\[3, 15\]](#) where each dimension of feature vector is considered independently, the context of relative magnitudes among different dimensions is implicitly and weakly encoded by [Eq. \(12.16\)](#).

With the PSIFT features represented by these binary signatures, the comparison between different features can be efficiently conducted by checking the Hamming distance between their binary signatures. Given the PSIFT query  $\mathbf{q} \in \mathcal{R}^d$ , we regard the candidate feature  $\mathbf{y} \in \mathcal{R}^d$  given by the SCH result of  $\mathbf{q}$  as a valid match if it satisfies the following criterion,

$$H(\mathbf{b}^{(\mathbf{q})}, \mathbf{b}^{(\mathbf{y})}) \leq \tau, \quad (12.17)$$

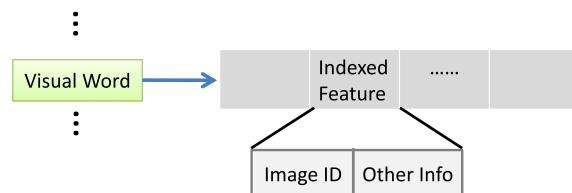
where  $\mathbf{b}^{(\mathbf{q})}$  and  $\mathbf{b}^{(\mathbf{y})}$  denote the binary signatures of  $\mathbf{q}$  and  $\mathbf{y}$ , respectively,  $H(\cdot, \cdot)$  denote the Hamming distance between two binary vectors, and  $\tau$  is a threshold.

## 5.12.5 Index strategy

Inspired by the success of the text-based information retrieval engines, inverted file structure [\[26\]](#) has been popularly used for large scale image search [\[1–3, 7, 8, 11, 12, 39\]](#). In essence, inverted file structure is a compact representation of a sparse matrix, where the row and the column denote visual word and image, respectively. In on-line retrieval, only those images sharing common visual words with the query image need to be checked. Therefore, the number of candidate images to be compared is greatly reduced, achieving efficient response.

### 5.12.5.1 Indexing with visual word

In the inverted file structure based on BoW model, each visual word is followed by an inverted list of entries. Each entry stores the ID of the image where the visual word appears, and some other clues for verification or similarity measurement, as illustrated in [Figure 12.11](#). For instance, Hamming Embedding [\[3\]](#) keeps a 64-bit Hamming code for each feature to verify the descriptor matching. Bundled Feature [\[11\]](#) stores the  $x$ -order and  $y$ -order of each SIFT feature located in the bundled area. The geometric clues, such as feature position, scale, and orientation, are also stored in the inverted file list for geometric consistency verification [\[2–4, 11, 12\]](#).



**FIGURE 12.11**

A toy example of image feature indexed with inverted file structure based on visual words.

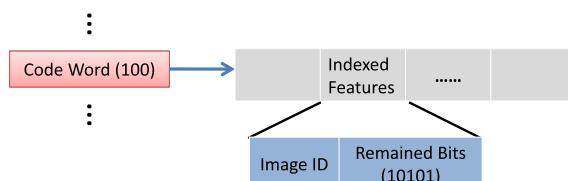
To further reduce the memory cost of inverted file structure, a visual word vector can be regarded as a global feature of an image and be mapped to a low-dimensional representation by a group of min-hash functions [10, 31, 53] or by a set of predefined sparse projection functions [32]. As a result, only a small constant amount of data per image need to be stored. Jegou et al. [54] proposed a different quantization strategy which jointly optimizes the dimension reduction and indexing to preserve the accuracy of feature vector comparison. The method aggregates all local descriptors of an image into a uniform and compact representation, which can achieve an excellent scalability in retrieval. However, it may not be able to handle partial-duplicate image search where the object of interest only occupies a small image region with a large area of cluttered background.

### 5.12.5.2 Indexing with binary SIFT

In image search, the problem of feature matching between images can be regarded as finding feature's nearest or approximately nearest neighbors within a certain range. When the feature amount becomes very large, say, over 1 billion, it is too computationally expensive to find the nearest neighbors by linearly comparing all features' binary vectors. When the feature space is Hamming space, we can generate code words from the binary feature so as to adapt to the inverted index structure for scalable retrieval. In the following, we take the binary SIFT discussed in [Section 5.12.3.3](#) for illustration. We first investigate how to define the code word. After that, we discuss a code word expansion scheme to include more candidate results for verification.

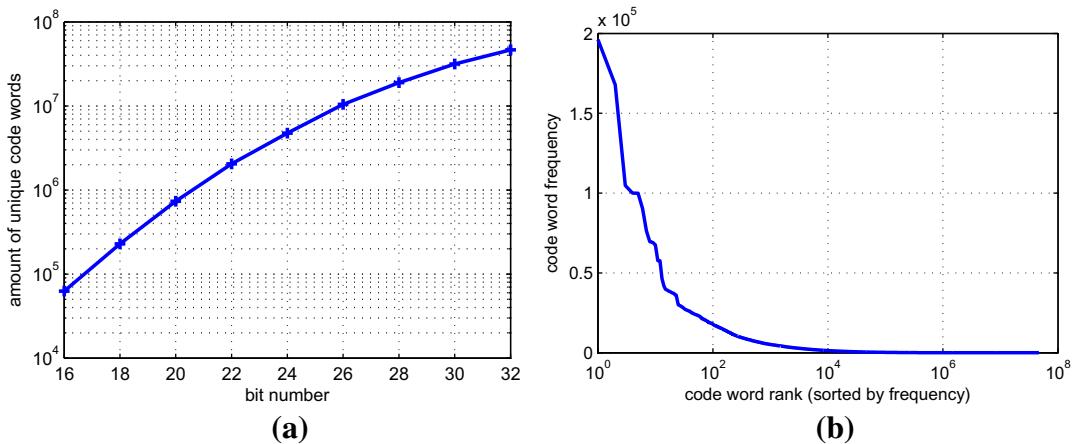
#### 5.12.5.2.1 Code word definition and exploration

In traditional inverted file structure for image search, a group of visual words are pretrained. And each visual word is followed with an entry list of image features, which are quantized to this followed visual word. Each indexed feature in the list records its image ID and some other clues. To adapt to the classic inverted file structure to index image features, we define *code word* by the first  $t$  bits of the binary SIFT. Then, the rest bits of features are recorded in the entry list of the corresponding code word for later verification. A toy example is shown in [Figure 12.12](#). Intuitively, if a code word is represented with  $t$  bits, the total number of code words could be amounted up to  $2^t$ . However, it is found from experiments that, when  $t$  increases to 20 and larger, the amount of nonempty code words becomes much smaller than  $2^t$ , as shown in [Figure 12.13a](#). For example, when  $t$  increases to 32, the total number of code words could be up to  $2^{32} \approx 4 \times 10^9$  (4 billion). However, the number of unique code words generated by scalar quantization (on 1 million image database) is even much less than  $10^8$ .



**FIGURE 12.12**

A toy example of image feature indexed with inverted file structure. The scalar quantization result of the indexed feature is an 8-bit vector (1001 0101). The first three bits denote its code word ID (100), and the remained 5 bits (10101) are stored in the inverted file list.



## FIGURE 12.13

(a) The amount of unique code words (top bits from 256-bit vector) for different on 1-million image database;  
 (b) frequency of code words among 1 million images before application of a stop-list.

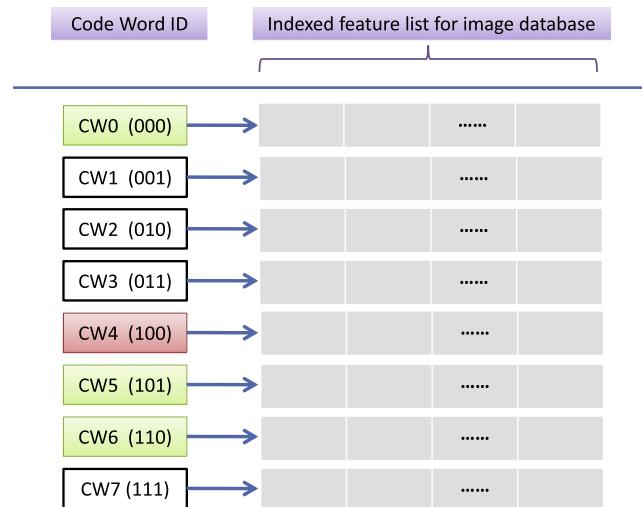
Generally, the more code words are generated, the shorter the average length of indexed feature list becomes, and the less the time cost is needed to query a new feature. However, we will introduce a soft quantization scheme ([Section 5.12.5.2.2](#)) to expand more code words for each query feature. And the number of expanded indexed feature lists is polynomial to  $t$ . To make a tradeoff, we select  $t = 32$ , and 46.5 million code words are obtained.

Figure 12.13b shows the distribution of code word occurrence on a 1-million-image database. It can be observed that, of the 46.5 million code words, only the top few thousand code words exhibit very high frequency. Those code words are prevalent in many images, and their distinctive power is weak. As suggested by [7], a stop-list can be applied to ignore those code words that frequently occur in the database images. Experiments reveal that a proper stop-list may not affect the search accuracy, but helps avoid checking many long inverted lists and achieve gain in efficiency.

Once all features of an image dataset have been indexed with the inverted file structure, given a new SIFT descriptor, it will be first quantized to a 256-bit vector with scalar quantization. Then through the top 32 bits, the corresponding code word can be located. And only the indexed features following the matched code word will be checked. Therefore, the searching space is greatly reduced. Finally, the exclusive-OR operation is performed between the remained 224 bits of the query vector and those of indexed features recorded in the entry list of the matched code word. A threshold  $\kappa$  on the Hamming distance between the 256-bit vectors needs to be set for true-match judgment, such that those matches with Hamming distance no larger than  $\kappa$  will be accepted as true matches.

### 5.12.5.2.2 Code word expansion

In Section 5.12.5.2.1, we define the code word by the top 32 bits of the BSIFT from scalar quantization. However, such simple processing will exclude some candidate features that have some flipping bits among the top 32 bits (e.g., 0 changes to 1) due to noise. To address this issue, a soft-decision strategy

**FIGURE 12.14**

A toy example of soft quantization with bit-stream code words. There are eight code words, each represented with a three-bit vector. Each code word is followed by an indexed image feature list.

can be applied to reduce such quantization loss. Assuming such flipping happens only to very few dimensions, features before and after the flipping should be still very similar, i.e., small Hamming distance. To identify these candidate features, it is desired to quickly enumerate all of its possible nearest neighbors (code words) within a predefined Hamming distance  $d$ , just by alternatively flipping some bits. This is equivalent to a tolerant expansion of the original code word.

As shown in the toy example in Figure 12.14, the code word of a new query feature is a bit-vector 100, i.e., CW4 in pink<sup>1</sup> color. To identify all of candidate features, its possible nearest neighbors (e.g., Hamming distance  $d = 1$ ) will be obtained by flipping 1 bit in turn, which generates three additional code words (in green color): CW0 (000), CW5 (101), and CW6 (110). These code words are the nearest neighbors of CW4 in the Hamming space. Then, besides CW4, the indexed feature lists of these three expanded code words will be also considered as candidate matches, and all features in these expanded lists will be further compared on their rest bit-codes.

## 5.12.6 Retrieval scoring

In multimedia retrieval, the target results in the index image database are assigned with a relevance score for ranking and returning to users. The relevance score can be defined either by measuring distance between image representation vectors or from the perspective of voting from relevant matches.

Based on the BoW model, an image is represented as a visual word vector by checking the appearance of visual words in the image. To distinguish the significance of visual words in different images, term

<sup>1</sup>For interpretation of color in Figure 12.14, the reader is referred to the web version of this book.

frequency (TF) and inverted document/image frequency (IDF) are widely applied in many existing state-of-the-art algorithms [1,3,7,9,11]. Generally, the visual word vectors weighted by TF and IDF are  $L_p$ -normalized for later distance computation.

In the on-line retrieval stage, after identifying those relevant images of a query by looking up the index table, it is necessary to determine the relevant score of those target images to the query image. Usually, the relevance score is defined by the  $L_p$ -normalized distance between the BoW vectors of the query and the database images, as defined in Eq. (12.18). When the codebook size is much larger than the local feature amount in images, the image vector by BoW is very sparse and we only need to check those visual words appearing in both images as illustrated in Eq. (12.19) [1], which is very efficient in practical implementation.

$$D(I_q, I_m) = \left( \sum_{i=1}^N |q_i - m_i|^p \right)^{\frac{1}{p}}, \quad (12.18)$$

$$D(I_q, I_m)^p = \sum_{i=1}^N |q_i - m_i|^p = 2 + \sum_{i|q_i \neq 0, m_i \neq 0} (|q_i - m_i|^p - q_i^p - m_i^p). \quad (12.19)$$

Some algorithms formulate the content-based retrieval as a voting problem. In [4,12,39], the relevance score is simply defined by counting how many pairs of local feature are matches across two images. In [11], the image similarity is defined as the sum of the TF-IDF score, which is further enhanced with a weighting term by matching bundled feature sets. In [55], contextual weighting is introduced to improve the classic vocabulary tree approach. Statistics of neighbor descriptors both on the vocabulary tree and in image spatial domain are efficiently incorporated.

Different from the above retrieval scoring methods, in [18], the retrieval scoring is formulated as a graph-based ranking problem. It builds a weighted undirected graph based on the retrieval results of one method. The graphs corresponding to multiple retrieval methods are fused to a single graph, based on which, link analysis [56] is conducted to identify the relevance score and rank the retrieval results.

## 5.12.7 Post-processing

The initially retrieval result list can be further refined by exploring the spatial context or enhancing the original query. Query expansion [8,57] and geometric verification [2–4,12,31] are two of the most successful post-processing techniques to boost the accuracy of large scale image search. In the following, we review those two schemes and introduce a geometric coding based verification scheme in detail [4].

### 5.12.7.1 Query expansion

Query expansion, leveraged from text retrieval, reissues the initially highly ranked results to generate new queries. Some relevant features, which are not present in the original query, can be used to enrich the original query to further improve the recall performance. In [8], to suppress the false positives that ruin the query expansion, strong spatial constraints between the query image and the candidate results are imposed, and a generative model is learned for the controllable construction of expanded queries.

Several expansion strategies, such as average query expansion, transitive closure expansion, recursive average query expansion, etc., are discussed.

In [57], two query expansion strategies are proposed to boost the retrieval quality. The first one is the intra-expansion which expands more target feature points similar to those in the query image. The second one, i.e., inter-expansion, explores those feature points that shall co-occur with the search targets but does not appear in the query image.

In [17], query expansion is formulated with a discriminative model. It takes spatially verified images as the positive training samples and images with low TF-IDF scores as the negative training samples. With those positive and negative BoW vectors of images, a linear SVM is trained, resulting in a weight vector. Finally, images are re-ranked by their distance from the SVM decision boundary.

### 5.12.7.2 Geometric verification by geometric coding

Spatial context is an important clue to remove false positive visual matches. The spatial context among local features of an image is critical in identifying duplicate image patches. After SIFT quantization, SIFT matches between two images can be obtained. However, due to quantization error and visual word ambiguity, the matching results are usually polluted by some false matches. Lots of work have been done on spatial verification. In [7], local spatial consistency is imposed to filter visual-word matches with low support. In [3], weak geometric consistency on scale and orientation is imposed to quickly filter potential false matches. In [2], global spatial verification is performed based on a variation [58] of RANSAC [59]. An affine model is estimated to filter local matches that fail to fit the model. Since full geometric verification with RANSAC [58,59] is computationally expensive, it is only used as a post-processing stage to process initially top-ranked candidate images. More efficient scheme to encode the spatial relationships of visual words is desired. Geometric coding is an alternative to ensure verification accuracy with small overhead in computation [4,60].

The key idea of geometric coding is to encode the geometric context of local SIFT features for spatial consistency verification. The geometric coding is composed of two types of coding strategies, i.e., geometric fan coding and geometric ring coding. The difference between the two strategies lies in the way how the image plane is divided according to an invariant reference feature point. Before encoding, the image plane has to be divided with a certain criterion that can address both rotation-invariance and scale-invariance. The criterion is designed via the intrinsic invariance merit of SIFT feature. We first introduce two geometric coding map generation schemes in [Section 5.12.7.2.1](#) and [Section 5.12.7.2.2](#), respectively. After that, we discuss how to perform geometric verification in [Section 5.12.7.2.3](#). In [Section 5.12.7.2.4](#), we discuss how to include those misremoved pairs with the enhancement based on affine estimation.

#### 5.12.7.2.1 Geometric fan coding

Geometric fan coding (GFC) encodes the relative positions between features in the axial direction in an image plane. A binary fan coding map  $F$  is generated to describe the relative position between each feature pair along the horizontal direction. For instance, given an image  $i$  with  $K$  features  $\{v_i\}$ , ( $i = 1, \dots, K$ ), its fan coding map  $F$  is defined as follows,

$$F(i, j) = \begin{cases} 0, & \text{if } v_j \text{ is on the left to } v_i, \\ 1, & \text{otherwise.} \end{cases} \quad (12.20)$$

In the fan coding map  $F$ , row  $i$  records other features' spatial relationships with feature  $v_i$  in the image. We can also interpret the map as follows. In row  $i$ , feature  $v_i$  is selected as the origin, and the image plane is uniformly divided into two parts along the vertical line. The coding map then shows in which side other features are located. Therefore, 1 bit either 0 or 1 can encode the relative spatial position of one feature to another.

The spatial context encoded by  $F$  in Eq. (12.20) is relatively loose. To describe the spatial relationship of local features more strictly, the geometric fan coding can be advanced to more general case. In other words, before comparing feature's relative positions along the horizontal line, we rotate the image plane counterclockwise by a series of  $r$  predefined angles  $\theta_k = \frac{k \cdot \pi}{r}$ , ( $k = 0, \dots, r - 1$ ) and then encode the new relative spatial relationship of local features for each rotation into a new coding map. After that, all coding maps are concatenated into a 3-D array  $GF$  as a general geometric fan coding map. For instance, after rotating the image plane counterclockwise by  $\theta_k$  according to the image origin point, the new location  $(x_i^{(k)}, y_i^{(k)})$  of feature  $v_i$  can be derived by its original location  $(x_i, y_i)$ ,

$$\begin{pmatrix} x_i^{(k)} \\ y_i^{(k)} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \quad (12.21)$$

Then, the generalized geometric fan coding map  $GF$  can be defined as follows,

$$GF(i, j, k) = \begin{cases} 0, & \text{if } v_j^{(k)} \text{ is on the left to } v_i^{(k)}, \\ 1, & \text{otherwise.} \end{cases} \quad (12.22)$$

The geometric fan coding map described above only captures invariance to translation and scaling change. Invariance to rotation change can be flexibly enhanced by considering the SIFT characteristic orientation when constructing the coding map. In other words, instead of checking whether a test feature  $v_j^{(k)}$  is on the left-hand side of the vertical line passing through a reference feature point  $v_i^{(k)}$ , we can check whether  $v_j^{(k)}$  is on the left-hand side of the line passing through  $v_i^{(k)}$  with normal direction exactly along the orientation of the reference feature point. With the above modification, the new coding maps will capture invariance to rotation, as well as translation and scaling change.

### 5.12.7.2.2 Geometric ring coding

Geometric ring coding (GRC) encodes the geometric context in radial direction of reference features. In GRC, with each SIFT feature as reference origin, the image plane is divided by rings or circles. A ring coding map  $R$  is constructed by checking whether other features are inside or outside of the ring.

$R$  describes whether other features are inside or outside of a ring defined by the reference feature. Given an image  $i$  with  $K$  features  $\{v_i\}$ , ( $i = 1, \dots, K$ ), its ring coding map  $R$  is defined as follows,

$$R(i, j) = \begin{cases} 1, & \text{if } d_{i,j} < s_i, \\ 0, & \text{otherwise,} \end{cases} \quad (12.23)$$

where  $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ ,  $s_i$  is the radius of ring and is proportional to SIFT scale of feature  $v_i$ :  $s_i = \alpha \cdot scl_i$ ,  $\alpha$  is a constant.

To describe the relative positions more strictly, we advance to general ring maps. For each feature,  $n$  concentric rings are drawn, with an equally incremental radius on the image plane. Then, the image

plane is divided into  $(n + 1)$  nonoverlapping circles. Correspondingly, according to the image plane division, a generalized geometric ring map  $GR$  is generated to encode the relative spatial positions of feature pairs.

$$GR(i, j) = \left\lfloor \frac{d_{i,j}}{s_i} \right\rfloor, \quad (12.24)$$

where  $d_{i,j}$  and  $s_i$  are the same as that in Eq. (12.23).

**Discussion:** The geometric coding schemes are a kind of quantization of the image plane. In this sense, it degenerates to model the local histograms as in spatial pyramid matching [61]. From the discussion above, it can be seen that, geometric coding can be very efficiently performed, but the whole geometric maps of all features in an image will cost considerable memory. Fortunately, there is no need to store these maps in the image search scenario. In fact, for each feature, we only need store the  $x$ - and  $y$ -coordinate, the characteristic scale, and the dominate orientation parameter of SIFT. When checking the feature matching of two images, only the coordinates and the characteristic orientations of these matched features are needed to generate the geometric coding maps for geometric verification in real time. The details about geometric verification are discussed in the next subsection.

### 5.12.7.2.3 Geometric verification with coding maps

In image search, after feature quantization, we can obtain the feature matching results between images. Denote that a query image  $I_q$  and a matched image  $I_m$  are found to share  $N$  pairs of matched SIFT features through vector quantization. Then the corresponding sub-geometric-fan-maps and sub-geometric-ring-maps of these matched features of both  $I_q$  and  $I_m$  can be generated and denoted as  $GF_q$  and  $GF_m$ ,  $GR_q$  and  $GR_m$ , respectively. If the query image and the matched image are partial-duplicate [4], it is likely that  $M$  out of  $N$  matches correspond to the duplicate patch and the spatial configuration of those  $M$  matched features in two images is very similar, with potential changes in translation, scale, or rotation. Since the geometric coding maps can handle those changes, the corresponding geometric fan coding maps of the  $M$  matched features in the two images, which are sub-maps of  $GF_q$  and  $GF_m$ , are identical. Similar observation also applies to the geometric ring coding maps. Therefore, the geometric verification problem can be formulated as identifying a sub-map corresponding to those spatially consistent matches from the complete geometric coding maps.

In the geometric verification, instead of directly identifying spatially consistent matches, we proceed to determine and remove those spatially inconsistent matches. After all those false matches are discarded, the remaining matches are the expected spatially consistent results. In fact, it is difficult to discover all spatially inconsistent matches once for all. However, it is possible to find one match that is most likely to be one of the false matches. Intuitively, such match will incur the most spatial inconsistency. Therefore, the key lies in how to formulate the spatial inconsistency.

To investigate the spatial inconsistency of matches, we perform logical Exclusive-OR (XOR) operation on  $GF_q$  and  $GF_m$ ,

$$V = GF_q \oplus GF_m. \quad (12.25)$$

Ideally, if all  $N$  matches are geometrically consistent,  $V$  are zero for all their entries. If some false matches exist, the entries of these false matches on  $GF_q$  and  $GF_m$  may be inconsistent. Those inconsistencies will cause the corresponding exclusive-OR result in  $V$  to be 1. In other words, the relative spatial consistency of feature  $j$  to the referred feature  $i$  is encoded with a  $r$ -bit

vector:  $C = [V(i, j, 1), \dots, V(i, j, r)]$ . And if feature  $i$  and feature  $j$  are geometrically inconsistent in two images, at least 1 bit in  $C$  will be “1.” We define the inconsistency from geometric fan coding as follows,

$$U_f(i, j) = \sum_{k=1}^r V(i, j, k). \quad (12.26)$$

The inconsistency from geometric ring coding is defined as:

$$U_r(i, j) = |GR_q(i, j) - GR_m(i, j)|. \quad (12.27)$$

We denote the overall geometric inconsistency as follows:

$$T(i, j) = \begin{cases} 1, & \text{if } U_f(i, j) > \tau \text{ or } U_r(i, j) > \beta, \\ 0, & \text{otherwise,} \end{cases} \quad (12.28)$$

where  $\tau$  and  $\beta$  are constant integers. When  $\tau$  or  $\beta$  is greater than zero,  $T$  in Eq. (12.28) can tolerate some drifting error of relative positions of local features.

Further, we quantitatively define the inconsistency of feature  $i$  with all other features as:

$$g(i) = \sum_{j=1}^N T(i, j). \quad (12.29)$$

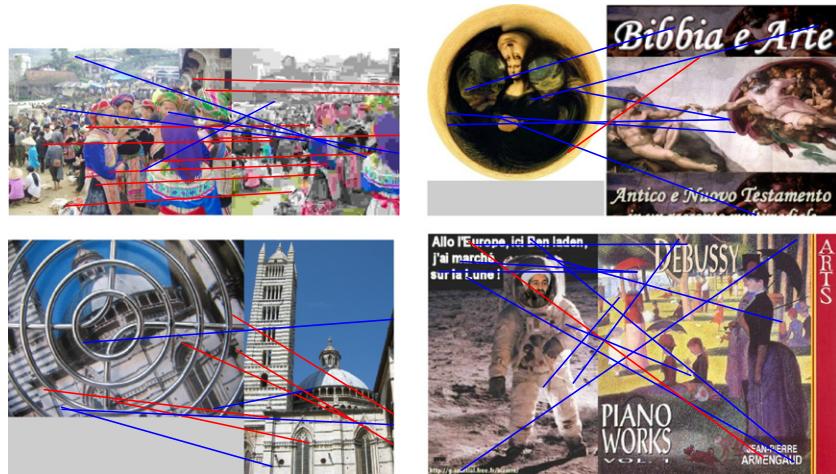
If there exists  $i$  with  $g(i) > 0$ , we define  $i^* = \operatorname{argmin}_i g(i)$ , then the  $i$ th pair will be most likely to be a false match and should be removed. Consequently, we can iteratively remove such mismatching pairs, until the maximal value of  $f$  is zero.

The computational complexity of geometric verification is  $O(r \cdot N^2)$ . It should be noted that in the iteration loop, when updating  $S$ , we just need to subtract the values of the inconsistent match, instead of recomputing it by Eq. (12.29).

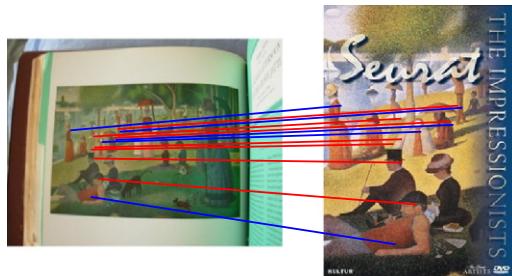
Figure 12.15 shows four real examples of the geometric verification results on two relevant and two irrelevant image pairs. All image pairs have many initial local matches after feature quantization. It can be observed that for those relevant partial-duplicate images, the geometric verification scheme can effectively identify and remove those spatially inconsistent matches. On the other hand, for those irrelevant image pairs, only very few, say one or two, local matches can pass the geometric verification scheme.

#### 5.12.7.2.4 Affine estimation for enhancement

Due to the unavoidable digital error in the detection of key points, some SIFT features exhibit some small drifting. With such drifting error, the relative location of features with the same  $x$ - or  $y$ -coordinate might be inverse, causing some true matches to fail to pass the geometric coding verification. Such phenomenon is prevalent in the case when two duplicate images share many matched feature pairs in a cluster. Besides, small affine transformation of images will exacerbate such error. Moreover, the error will also be worsened by large geometric coding factor  $r$  and  $n$ . As illustrated in Figure 12.16, although the duplicate image pair shares 12 true matches, 5 of them are discovered as false matches and fail to pass the geometric coding based verification. In this section, we discuss how to address such SIFT drifting problem by affine estimation based on matched pairs passing the geometric verification.

**FIGURE 12.15**

Real examples of geometric verification results with geometric coding on relevant (left) and irrelevant (right) image pairs. Each line denotes a local match across two images, with the line endpoint at the key point location of the corresponding SIFT feature. The red lines denote those local matches passing the geometric verification while the blue lines denote those that fail to pass the verification. The initial local matches are obtained with a visual codebook quantizer containing 1 million visual words. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this book.)

**FIGURE 12.16**

An instance of matching error due to SIFT drifting. The red lines denote the true matches that pass the geometric coding based verification, while the blue lines denote those that fail. (Best viewed in color PDF). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this book.)

Since still many true matches remained, the matched images will be assigned a comparatively high similarity score (defined as the number of local matches passing the geometric verification) and consequently the effect of those false negatives on the retrieval performance is very small. In fact, we can also discover those false negatives by means of those discovered spatial-consistent matches. In other

words, we can estimate the affine transformation from target image to query image. Generally, there are six parameters of an affine transformation. Let  $(u, v)$  and  $(x, y)$  be the coordinates of the matched feature points in matched image and query image, respectively. Then the affine transformation between  $(u, v)$  and  $(x, y)$  can be expressed as Eq. (12.30).

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} m_1 & m_2 \\ m_3 & m_4 \end{pmatrix} \cdot \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}. \quad (12.30)$$

The least-squares solution for the parameters  $\mathbf{p} = (m_1, m_2, m_3, m_4, t_1, t_2)$  can be determined by solving a corresponding normal equation [6]. After estimating the parameters of the affine transformation, we can check the other matching pairs that fail to pass the geometric verification. In details, assume  $(\hat{x}, \hat{y})$  is the feature point in the query image estimated by the transformation model. Then the mapping error of the matching feature point pair is

$$e = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}. \quad (12.31)$$

To check the validness of the matching, we set a threshold  $T$  such that, if the mapping error of a matched pair that fails to pass the geometric verification is less than  $T$ , we will consider that pair as a true match. Since the duplicated images may undergo scale changes, it is reasonable to weight the threshold with a factor reflecting the scale difference. Consequently,  $T$  is defined as

$$T = s \cdot \tau, \quad (12.32)$$

where  $\tau$  is a weighting constant, and  $s$  is the average scale ratio between all true positive matches, defined as,

$$s = \frac{1}{M} \cdot \sum_{i=1}^M \frac{scl_i^q}{scl_i^l}, \quad (12.33)$$

where  $scl_i^q$  and  $scl_i^l$  are the scale value of the  $i$ th truly matched SIFT features from the matched image and the query image, respectively.

Finally, false negative matching pairs due to SIFT drifting will be identified and kept. It should be noted that, unlike RANSAC [59, 58], the affine model estimation is performed only once. Therefore, it is very efficient in implementation. With the affine estimation for enhancement, we can effectively and efficiently recover those false negative matches failing to pass the geometric verification due to SIFT drifting.

### 5.12.8 Conclusion

In this chapter, we have investigated the complete general framework on multimedia content-based visual retrieval. We focus on the five key modules of the general framework, i.e., feature extraction, feature quantization, image indexing, retrieval scoring, and post-processing. For each component, we have discussed the key problems and introduced a variety of representative strategies and methods.

Despite the advance in multimedia content-based visual retrieval, there is still gap toward semantic-aware retrieval from visual content. Due to the tremendous diversity and quantity in multimedia visual data, most existing methods are unsupervised. To proceed toward semantic-aware retrieval, scalable supervised or semi-supervised learning methods are promising to boost the content-based retrieval quality. The advance of scalable machine learning methods is anticipated with great potential in multimedia content-based visual retrieval.

---

## Acknowledgment

Wengang Zhou, Houqiang Li, and Qi Tian would like to thank their co-authors Yijuan Lu, Ming Yang, Meng Wang, et al. for their contributions to the related work that is covered in this chapter. This work was supported in part to Dr. Zhou by the Fundamental Research Funds for the Central Universities under contract No. WK2100060014 and the start-up funding from the University of Science and Technology of China under contract No. KY2100000036, in part to Dr. Li by NSFC under contract No. 61325009, No. 61390514, and No. 61272316, and in part to Dr. Tian by ARO grant W911NF-12-1- 0057, Faculty Research Awards by NEC Laboratories of America, Google, and FXPAL, and 2012 UTSA START-R Research Award, respectively. This work was supported in part by National Science Foundation of China (NSFC) under contract No. 61128007.

---

## References

- [1] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.
- [2] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
- [3] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 304–317.
- [4] W. Zhou, H. Li, Y. Lu, Q. Tian, Large scale image search with geometric coding, in: Proceedings of the ACM International Conference on Multimedia, 2011, pp. 1349–1352.
- [5] Copydays\_dataset, <<http://lear.inrialpes.fr/~jegou/data.php>>.
- [6] D.G. Lowe, Distinctive image features from scale invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
- [7] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2003, pp. 1470–1477.
- [8] O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman, Total recall: automatic query expansion with a generative feature model for object retrieval, in: Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.
- [9] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: improving particular object retrieval in large scale image databases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
- [10] O. Chum, J. Philbin, A. Zisserman, Near duplicate image detection: min-hash and tf-idf weighting, Proc. BMVC, vol. 3, 2008, p. 4.
- [11] Z. Wu, Q. Ke, M. Isard, J. Sun, Bundling features for large scale partial-duplicate web image search, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 25–32.
- [12] W. Zhou, Y. Lu, H. Li, Y. Song, Q. Tian, Spatial coding for large scale partial-duplicate web image search, in: Proceedings of the ACM International Conference on Multimedia, 2010, pp. 511–520.

- [13] O. Chum, A. Mikulik, M. Perdoch, J. Matas, Total recall II: query expansion revisited, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 889–896.
- [14] Y. Zhang, Z. Jia, T. Chen, Image retrieval with geometry-preserving visual phrases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 809–816.
- [15] X. Zhang, L. Zhang, H. Shum, QsRank: query-sensitive hash code ranking for efficient  $\epsilon$ -neighbor search, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2058–2065.
- [16] J. He, J. Feng, X. Liu, T. Cheng, T. Lin, H. Chung, S. Chang, Mobile product search with bag of hash bits and boundary reranking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3005–3012.
- [17] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2911–2918.
- [18] S. Zhang, M. Yang, T. Cour, K. Yu, D.N. Metaxas, Query specific fusion for image retrieval, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 660–673.
- [19] Q. Tian, S. Zhang, W. Zhou, R. Ji, B. Ni, N. Sebe, Building descriptive and discriminative visual codebook for large-scale image applications, *Multimedia Tools Appl.* 51 (2) (2011) 441–477.
- [20] W. Zhou, H. Li, Y. Lu, Q. Tian, Large scale partial-duplicate image retrieval with bi-space quantization and geometric consistency, in: Proceedings of the IEEE International Conference Acoustics Speech and Signal Processing, 2010, pp. 2394–2397.
- [21] S. Zhang, Q. Tian, G. Hua, Q. Huang, S. Li, Descriptive visual words and visual phrases for image applications, in: Proceedings of the ACM International Conference on Multimedia, 2009, pp. 75–84.
- [22] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, Q. Tian, Building contextual visual vocabulary for large-scale image applications, in: Proceedings of the ACM International Conference on Multimedia, 2010, pp. 501–510.
- [23] W. Zhou, Q. Tian, Y. Lu, L. Yang, H. Li, Latent visual context learning for web image applications, *Pattern Recognit.* 44 (10) (2011) 2263–2273.
- [24] Tineye\_Image\_Search, <<http://www.Tineye.com/>>.
- [25] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, *Int. J. Comput. Vis.* 60 (1) (2004) 63–86.
- [26] R. Baeza-Yates, B. Ribeiro-Neto, et al., *Modern Information Retrieval*, vol. 463, ACM press, New York, 1999.
- [27] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, *Image Vis. Comput.* 22 (10) (2004) 761–767.
- [28] H. Bay, T. Tuytelaars, L. Van Gool, Surf: speeded up robust features, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 404–417.
- [29] P.F. Alcantarilla, A. Bartoli, A.J. Davison, Kaze features, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 214–227.
- [30] S. Zhang, Q. Tian, K. Lu, Q. Huang, W. Gao, Edge-SIFT: discriminative binary descriptor for scalable partial-duplicate mobile search, *IEEE Trans. Image Process.* 22 (7) (2013) 2889–2902.
- [31] O. Chum, M. Perdoch, J. Matas, Geometric min-hashing: finding a (thick) needle in a haystack, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 17–24.
- [32] H. Jégou, M. Douze, C. Schmid, Packing bag-of-features, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2009, pp. 2357–2364.
- [33] L. Zheng, S. Wang, Z. Liu, Q. Tian, Lp-norm idf for large scale image search, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.
- [34] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: binary robust independent elementary features, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 778–792.
- [35] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: an efficient alternative to sift or surf, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2564–2571.

- [36] A. Alahi, R. Ortiz, P. Vandergheynst, Freak: fast retina keypoint, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 510–517.
- [37] S. StefanLeutenegger, M. Chli, R. Siegwart, Brisk: binary robust invariant scalable keypoints, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2548–2555.
- [38] E. Rosten, R. Porter, T. Drummond, Faster and better: a machine learning approach to corner detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (1) (2010) 105–119.
- [39] W. Zhou, Y. Lu, H. Li, Q. Tian, Scalar quantization for large scale image search, in: Proceedings of the ACM International Conference on Multimedia, 2012, pp. 169–178.
- [40] T. Tuytelaars, C. Schmid, Vector quantizing feature space with a regular lattice, in: Proceedings of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.
- [41] J. Bentley, K-d trees for semidynamic point sets, in: Proceedings of the Sixth Annual Symposium on Computational Geometry, 1990, pp. 187–197.
- [42] C. Silpa-Anan, R. Hartley, Localization using an image map, in: Proceedings of the Australian Conference on Robotics and Automation, 2004.
- [43] M. Jain, H. Jégou, P. Gros, Asymmetric hamming embedding: taking the best of our bits for large scale image search, in: Proceedings of the ACM International Conference on Multimedia, 2011, pp. 1441–1444.
- [44] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: Proceedings of the IEEE Symposium on Foundations of Computer Science, 2006, pp. 459–468.
- [45] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 2130–2137.
- [46] J. Wang, S. Kumar, S. Chang, Semi-supervised hashing for scalable image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3424–3431.
- [47] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Neural Information Processing Systems, 2008, pp. 3424–3431.
- [48] Y. Gong, S. Lazebnik, Iterative quantization: a procrustean approach to learning binary codes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 817–824.
- [49] W. Zhou, H. Li, Y. Lu, M. Wang, Q. Tian, Visual word expansion and BSIFT verification for large-scale image search, *Multimedia Syst.* (2013) 1–10.
- [50] S. Arya, D.M. Ann, Ann: Library for Approximate Nearest Neighbor Searching, <<http://www.cs.umd.edu/~mount/ANN/>>.
- [51] Y. Ke, R. Sukthankar, Pca-sift: a more distinctive representation for local image descriptors, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 506–513.
- [52] P. Viola, M. Jones, Robust real-time face detection, *Int. J. comput. Vis.* 57 (2) (2004) 137–154.
- [53] O. Chum, J. Philbin, M. Isard, A. Zisserman, Scalable near identical image and shot detection, in: Proceedings of the ACM International Conference on Image and Video Retrieval, 2007, pp. 549–556.
- [54] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3304–3311.
- [55] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, T. Han, Contextual weighting for vocabulary tree based image retrieval, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 209–216.
- [56] L. Page, S. Brin, R. Motwani, T. Winograd, The Page rank Citation Ranking : Bringing Order to the Web, Stanford InfoLab, 1999.
- [57] Y. Kuo, K. Chen, C. Chiang, W. Hsu, Query expansion for hash-based image object retrieval, in: Proceedings of the ACM International Conference on Multimedia, 2009, pp. 65–74.
- [58] O. Chum, J. Matas, Matching with prosac-progressive sample consensus, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 220–226.
- [59] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* 24 (6) (1981) 381–395.

- [60] W. Zhou, H. Li, Y. Lu, Q. Tian, SIFT match verification by geometric coding for large-scale partial-duplicate web image search, *ACM Trans. Multimedia Comput. Commun. Appl.* 9 (1) (2013) 4.
- [61] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

# Joint Audio-Visual Processing for Video Copy Detection

# 13

Zhu Liu<sup>\*</sup>, Eric Zavesky<sup>†</sup>, David Gibbon<sup>\*</sup>, and Behzad Shahraray<sup>\*</sup>

<sup>\*</sup>200 South Laurel Avenue, Middletown, NJ 07748, United States

<sup>†</sup>9505 Arboretum Blvd, Austin, TX 78759, United States

## 5.13.1 Introduction

At the core of every computer vision task is a system that accesses prior knowledge. While video copy detection may not be the answer for all of these tasks, indexing and linking to previous examples facilitates context discovery and a wealth of powerful applications. Previous methods created to index and understand content are largely generalized or parameterized, losing specific information about any one piece of content. The increasing rate of content creation, unlikely to slow or reverse, demands that these general indexing systems be augmented to quickly and uniquely identify content, whether it is recaptured via a sensor (i.e., camera or microphone) or discovered digitally after various content transformations (i.e., an edited mash-up, overlaid with text, video, or other audio).

Video copy detection is essential for many applications, for example, discovering copyright infringement of multimedia content, monitoring commercial air time in TV broadcast, automatically labeling the music in radio broadcast, metadata extraction, querying audio/video by example, etc. Generally there are two complementary approaches for video copy detection: digital video watermarking and content-based copy detection. The first approach refers to the process of embedding irreversible information in the original audio and visual streams, where the watermarking can be either perceptible or imperceptible. The second approach extracts content-based features directly from the audio and visual streams, and uses these features to determine whether one video is a copy of another. No change needs to be made in the original media in this approach. In this chapter, the focus is on the second approach.

With video content search techniques maturing in recent years, applications and services were built to improve the users' experience in video consumption, especially on mobile platforms [1]. In the image domain, SnapTell (part of Amazon) finds products like books, DVDs, CDs, or video games using a picture of its cover without relying on UPC/EAN barcode images. In the audio domain, Shazam identifies music and advertisements using 5–7 s segments. Other services like IntoNow, Audible Magic, SoundHound, Zeitera, Kooaba, and Civolution leverage live audio or video copy detection for automated content recognition and immediate retrieval. In all of these services, extra metadata, including song name, artist, album, link to purchase, etc., are available once the content has been identified. These linking services improve the user experience and enable social interaction such as posting comments and preferences on YouTube.

Video copy detection has become a very active topic in the last decade, originating from single modality approaches using either audio- or visual-based methods exclusively. The visual-based method benefits from the research achievements in multiple areas, including content-based video analysis [2], computer vision, machine learning [3], etc. The main efforts have been focused on identifying the effective visual features and visual fingerprint to represent the content of individual image, devising the efficient way to index and search big dataset of images, and employing the right machine learning tools to incorporate the temporal spatial variations of videos and improve the overall detection performance. In [4], a set of visual-based video copy detection techniques using different descriptors and voting functions were tested and compared within the same evaluating framework. More recent work extending these methods can be found in [5–8].

The audio-based methods adopted many technologies developed in the classic speech signal processing [9], general audio content analysis [10], audio event categorization, music identification and search [11,12], as well as machine learning. Similar to the visual counterpart, most research work in audio-based approaches is on investigating and devising discriminative acoustic features and fingerprint to represent the audio content compactly. Compared to the huge amount of visual data processed in visual-based methods, the audio-based approaches tackle the 1D audio input and its main advantage is intrinsic efficiency. Given the high accuracy of state-of-the-art automatic speech recognition (ASR) engines, the linguistic information extracted from audio track may also provide useful high-level semantics that help the copy detection task. While most audio-based copy detection methods use feature-level information instead of recognized textual information, recent work extending these methods is reported in [13,14].

Multimodal information fusion has been proven effective and widely adopted in many related areas, including video content analysis, video event detection, video summarization, video search, and data mining [15,33]. The main topic of contention is how to fuse the complementary information from multiple sources to improve the system accuracy and robustness performance. Atrey et al. surveyed the state-of-the-art fusion strategies for multimedia analysis, and reported the basic concept, advantages, weakness, and applications of various fusion methods in [16]. While visual-based copy detection methods usually perform better than their audio-only counterpart, combining cues from both audio and visual streams boosted the video copy detection results. Generally speaking there are two types of fusion schemes: early fusion (feature level) and late fusion (decision level). In the first category, raw information from the two modalities (for example, the low level or certain aggregated high-level audio and visual features) is merged first and then the decision module detects the video copies based on the combined information. In the second category, audio- and visual-based video copy detectors work independently first, and then the two intermediate decision results are combined into the final result. Theoretically, the early fusion scheme does not lose information before the final decision is made while the other scheme may lose certain mutual information at an early stage. But in practical implementations, the late fusion scheme yields better performance and is more popular as the three main modules (audio- and visual-based detectors as well as the fusion module) can be independently designed and optimized.

Research in video copy detection touches many fields but one community-accepted evaluation task is TRECVID (TREC Video Retrieval Evaluation) [17], sponsored by the National Institute of Standards and Technology (NIST). Its goal is to encourage research through laboratory-style video evaluation tasks like shot boundary detection (SBD), segmentation, summarization, content-based copy detection (CCD), multimedia event detection, semantic indexing, instance-based search (INS), etc. Looking specifically at the last TRECVID evaluation of the CCD task (TRECVID 2011), systems by Peking University

(PKU), the Computer Research Institute of Montreal (CRIM), and the National Institute for Research in Computer Science and Control (INRIA) were among the top performers. PKU's system [18] combines three detectors based on audio, local visual features, and global visual features in a cascade architecture, and adopts a temporal pyramid matching method to aggregate frame into video level results. CRIM [19] relies on temporally normalized visual features extracted from 16 subsquares in the RGB color space, and employs nearest-neighbor fingerprints for high accuracy search. The same search method is applied on Mel-frequency cepstral coefficient audio features. Visual features in INRIA's system [20,21] are center-symmetric local binary patterns, and audio features are log-energies of bandpass filters with an audio-video early fusion method.

This chapter describes algorithms for a joint visual and audio copy detection system and presents several applications enabled by such systems, where the goal is to link (detect and identify) a new piece of content to a known piece of content. Underlying algorithms were designed to accommodate a number of complex audio and video transformations and their efficacy was analyzed through international evaluations. The remainder of this chapter describes related work, the algorithms in detail, and a number of consumer and commercial applications. The main contribution of this work is a robust, large-scale copy detection system using computer vision and efficient indexing exemplified through interesting and useful modern application prototypes. Enhancements from the existing system reported in [22] include an audio-based copy detection module, an effective late fusion mechanism, and the successful prototyping of consumer applications. Results from international evaluations also demonstrate that the video copy detection engine is highly efficient and robust.

This chapter is organized as follows. [Section 5.13.2](#) presents a review on the visual-based video copy detection methods, and the audio-based counterpart is described in [Section 5.13.3](#). Then different schemes for fusing audio- and visual-based subsystems are elaborated in [Section 5.13.4](#). In [Section 5.13.5](#), experimental results on video copy detection are discussed and a prototype application built on the algorithm is demonstrated. In [Section 5.13.6](#), the object-level video search system is evaluated and a mobile app for instance search is shown. Finally, conclusions are drawn while listing a few potential future research directions in [Section 5.13.7](#).

## 5.13.2 Visual-based video copy detection algorithm

This section details core analysis methods to efficiently segment, represent, index, and retrieve linked pieces of content based on visual information.

### 5.13.2.1 Overview

[Figure 13.1](#) illustrates the block diagram of a general visual-based video copy detection algorithm. The processing of reference videos is as follows. A shot boundary detection (SBD) algorithm is applied to segment the video into shots with homogeneous content and one keyframe is selected for each shot. The aim of shot boundary detection is data reduction, since indexing every frame in reference videos is computationally intractable. To proactively cope with the transformations in the query video, it is beneficial to normalize the original reference keyframe with certain transformations, for example, half resolution and strong re-encoding. These two additional versions of the reference keyframe are useful for detecting query keyframes with Picture-in-picture (PiP) and strong re-encoding transformations,

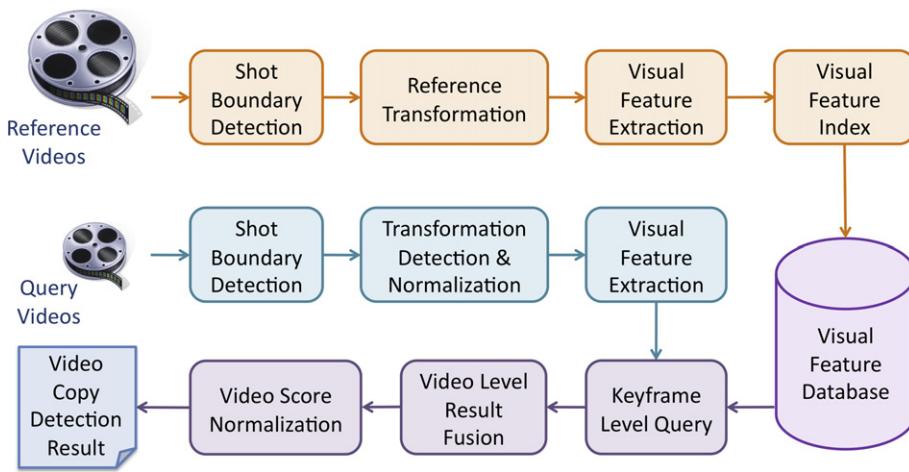
**FIGURE 13.1**

Diagram of the visual-based video copy detection algorithm.

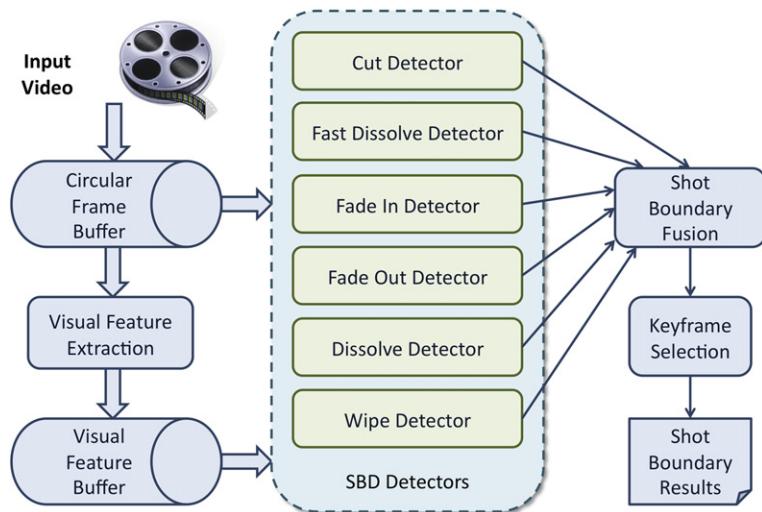
and they can be processed independently in parallel with the original keyframes. Then, visual features are extracted from the keyframes to represent the image content, and they are indexed and stored in the visual feature database for efficient indexing and query.

For each query video, the same shot boundary detection algorithm is also employed to segment the video and extract keyframes. It is assumed that the copied content in the query usually has been through some video transformations, including video stretch, PiP, scale change, shift, etc. For some common transforms, it is useful to detect them and reverse the transformation effect. For example, the stretched video is rescaled to the original resolution and the embedded picture-in-picture region is rescaled to half original resolution. Similar to the reference video processing, the original keyframe and the normalized keyframes pass through the following processing steps independently. After visual features are extracted from the keyframes, they are used as queries to identify the similar reference keyframes. This is the key step since it serves as the initial results for the postprocessing and strongly determines the performance of the final result. Variety of tricks to improve the query efficiency and effectiveness will be discussed in detail in the following sections. Next, for each query keyframe, the query results from different sets (the original and the transformed versions) are merged based on their relevance scores. The keyframe level query results are then combined in the video level result fusion module, where the temporal relationship among the keyframe query results as well as their relevance scores are considered to determine the best matching video chunks. Finally, the match scores can be normalized based on the requirements of different applications, and the overall visual-based copy detection results are generated.

In the following subsections, details of each component will be described.

### 5.13.2.2 Shot boundary detection

Shot boundary detection has been widely studied for the last two decades. Some of the early works can be found in [23–25, 15]. More recent works have been reported in the TRECVID shot boundary

**FIGURE 13.2**

Shot boundary detection algorithm.

detection (SBD) evaluation task [26,27]. Some of the top performing systems adopted a divide and conquer strategy, and focused on fusing independent detectors. Evaluation results showed that systems usually performed well on detecting abrupt changes (cuts), but the gradual transitions still posed a challenge. This section describes AT&T's SBD system shown in Figure 13.2 [28].

There are three main components in this SBD system: visual feature extraction, shot boundary detectors, and result fusion. The framework is flexible to incorporate different kinds of detectors, and this figure shows six detectors, targeting the six most common types of shot boundaries. They are cut, fast dissolve (last less than five frames), fade in, fade out, dissolve, and wipe. Depending on the applications, additional detectors, for example, subshot and motion (including panning, zooming, tilting, etc.) may be included. Essentially, each detector is a finite state machine (FSM), which has different numbers of states to detect the target transition pattern and locates the transition boundaries. Support vector machines (SVM) based transition verification method is applied in the cut, fast dissolve, and dissolve detectors. The results of all detectors are then fused together based on a predetermined priority scheme.

The FSMs of all detectors depend on two types of visual features: intra-frame and inter-frame features. The intra-frame features are extracted from a single frame, which include color histogram, edge, and their statistics. The inter-frame features rely on the current frame and one previous frame, and they capture the motion compensated intensity matching errors and histogram changes. To select a representative frame (keyframe) for each shot, a score is computed for each frame based on its sharpness (to avoid blur introduced by both motion and focus) and the number of detected faces (optional due to its high computational cost), and the frame with the top score is chosen as the keyframe. Since the computational cost of this keyframe selection method is high, a simpler method, for example, using the first frame of each shot as the keyframe is usually adopted in real-time applications.

### 5.13.2.3 Transformation detection and normalization

In copied content, different transformations can be applied intentionally (e.g., to elude the copyright infringement detection) or unintentionally (e.g., to convert it to the right format for streaming). Typical transformations include camcording, picture-in-picture (PiP), insertions of pattern, strong re-encoding, change of gamma, decrease in quality, and combinations of individual transforms. It is not realistic to detect all kinds of transformations, and recover the effect, since some of them are irreversible. In this system, focus was given to letterbox detection and picture-in-picture detection. Robust edge features detect the bounding rectangles for both letterbox and PiP regions, and the detected regions are resized to original resolution for letterbox and half original resolution for PiP.

#### 5.13.2.3.1 Letterbox detection

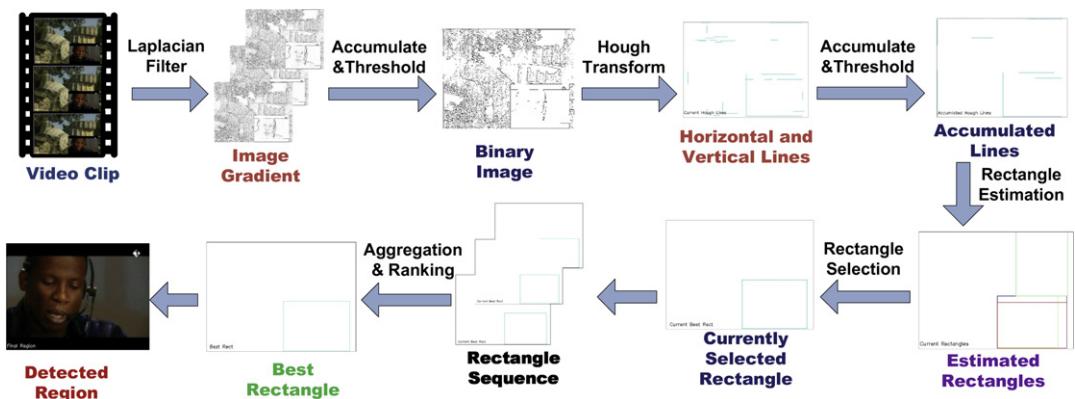
Both edge information and the temporal intensity variance of each pixel are utilized to detect letterbox. For each frame, the Canny edge detection algorithm is applied to locate all edge pixels. Then based on the edge direction, a horizontal edge image and a vertical edge image are extracted. To cope with the noise in the original video, the edge images have to be smoothed. The smoothing process includes removing the short edge segments, and merging adjacent edge segments that are less than 5 pixels away. After all frames of the query video are processed, the mean horizontal (and vertical) edge image is computed for the entire video and it is projected horizontally (and vertically) for computing the horizontal (and vertical) edge profile. For stretched, cropped, or shifted videos, there are significant peaks in the horizontal and/or vertical edge profiles. The algorithm identifies the maximum horizontal (and vertical) peak near the boundary, such that it is within one eighth of the height (and the width). Usually, these detected profile peaks indicate the letterbox boundaries.

When the video is too noisy, profile peaks may not be detected reliably. When these conditions are detected, a fallback option uses the intensity variation of each pixel across the entire video. For the embedded video region, the intensity variance of each video is relatively large due to the dynamic video content. For the letterbox region, the variance is much smaller. Comparing the pixel intensity variance value to a preset threshold, each pixel is classified into either the video pixel or the non-video pixel. Letterbox boundaries are set such that the percentage of video pixels in the letterbox region is less than 1%.

#### 5.13.2.3.2 Picture-in-picture (PiP) detection

The PiP detection method is essentially based on the observation that the PiP region boundary consistently appears across the whole video. Consideration of the PiP region properties, like geometric constraints, produces a detection method as illustrated in [Figure 13.3](#). Given a video of  $N$  frames, at frame  $f$ , the method detects the PiP region based on all frames from the video's start to frame  $f$ . The final result is determined by aggregating the detected regions for the entire video. The following steps are used to detect the PiP region in the current frame.

- Step 1: The image gradient of current frame  $f$  (gray image of that frame) is calculated by using the Laplacian filter.
- Step 2: The gradients of frames 0 to  $f$  are accumulated and then averaged to form an average gradient image  $G$  (a matrix), which is converted into a binary mask  $B$  using Otsu's algorithm.
- Step 3: The Hough transform is applied on the binary image  $B$  to achieve an edge image  $E$  of vertical and horizontal edges. The Hough transform is adopted in this context with the aim of removing

**FIGURE 13.3**

Overview of the picture-in-picture detection method.

noisy edges. In addition, an edge smoothing procedure is applied to remove some particular short and long edges.

- Step 4: A similar process as Step 2 is employed to produce a more robust edge image  $B'$ .
- Step 5: Based on the edge image  $B'$ , multiple candidate rectangles (a PiP region is bounded by a rectangle) are estimated and assigned scores by considering the geometric constraints, including the aspect ratio and the size of a rectangle, etc. To remove invalid rectangles, the following criteria are used.
  - Distance constraint: Parallel edges of the rectangle should not be too close or far away.
  - Location constraint: The vertical edges should be located between the two horizontal edges, and the horizontal edges should be located between the two vertical edges. Crossing constraint: The extended lines of two orthogonal edges of a rectangle cannot cross at the middle of each edge. [Figure 13.4](#) illustrates this constraint. Given a horizontal edge (the solid horizontal line in [Figure 13.4](#)), the vertical edges should appear at the valid regions (the gray regions in [Figure 13.4](#)).

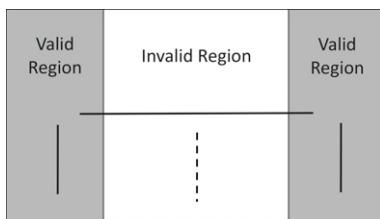
**FIGURE 13.4**

Illustration of the crossing constraint. Given a horizontal edge, the gray regions are valid for a vertical edge (e.g., the solid vertical line) to appear. The valid region only considers crossing constraints for the current horizontal edge.

For instance, while the solid vertical lines are valid vertical edges given the horizontal edge, the dotted vertical line is not a valid one.

After removing the invalid rectangles, a valid candidate rectangle set  $R = \{R_i\}, i = 1, \dots, M$  is obtained, where  $M$  is the cardinality of the set.

- Step 6: For each rectangle  $R_i$ , the system computes a score based on its aspect ratio and size and then chooses the rectangle with the highest value as the detected PiP region for the current frame. The score of rectangle  $R_i$  is calculated as,

$$S(R_i) = S_A(R_i) \times S_S(R_i) \times E(R_i),$$

where  $S_A(R_i)$  and  $S_S(R_i)$  are the scores of  $R_i$  based on its aspect ratio and size, respectively, and  $E(R_i)$  is the number of detected edges for  $R_i$ .  $S_A(R_i)$  is computed by the agreement between the aspect ratio of  $R_i$  and that of the background frame  $F$ ,

$$S_A(R_i) = \exp\{-|AR(R_i) - AR(F)|\},$$

where  $AR(\cdot)$  is the aspect ratio of a rectangle (width/height).  $S_S(R_i)$  is computed as,

$$S_S(R_i) = \begin{cases} 1, & \theta_1 \leq \frac{\text{size}(R_i)}{\text{size}(F)} \leq \theta_2, \\ 0, & \text{otherwise,} \end{cases}$$

where  $\text{size}(\cdot)$  is the area of a rectangle, and  $\theta_1$  and  $\theta_2$  are two predefined thresholds.

At each frame, these six steps estimate one PiP region and after processing the whole video, a set of detected regions  $P$  is produced. For a candidate region  $C_i$ , a score is assigned based on two strategies. One strategy is by voting, and the other is by summation of scores. In both cases, the top ranked detected region is chosen as the final PiP result.

- Voting: The score of each candidate region  $C_i$  is the number of frames within which  $C_i$  is detected as a PiP region.
- Summation: The summing strategy uses the summation of scores of  $C_i$  in each frame within which it is detected as a PiP region as the final score of region  $C_i$ .

#### 5.13.2.4 Visual feature extraction

In general, there are two categories of visual features: global features and local features. The features from the first category are extracted based on the entire image, and those from the second category are computed based on a local region. The following two subsections will focus on these two categories.

##### 5.13.2.4.1 Global visual features

There are varieties of global visual features, for example the histogram of intensity values, etc. In [3], three categories of global visual features are reported, which are color, shape, and texture features. A more recent addition is the GIST descriptor proposed in [29]. The main idea is that the holistic spatial scene properties can be reliably estimated using spectral and coarsely localized information. Specifically, the orientation histograms extracted from grids are used to represent the perceptual dimensions,

including naturalness, openness, roughness, expansion, and ruggedness. In [30], the GIST feature was evaluated for the near-duplicate detection task, and it provided comparable accuracy as the state-of-the-art local feature approaches with higher efficiency and less memory footprint.

This section presents three types of global features that are proven to be effective for near-duplicate detection tasks.

### Color moments

Color moments represent the distribution of each color channel in an image by mean, standard deviation, and the third root of the skewness [31]. Assuming the  $i$ th channel of the input image  $I$  is denoted by  $I^i$ , and the resolution of the image is  $M \times N$  where  $M$  is the number of rows and  $N$  is the number of columns. The color moments can be computed by the following equations:

$$E_i = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I^i(x, y),$$

$$\sigma_i = \left\{ \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [I^i(x, y) - E_i]^2 \right\}^{\frac{1}{2}},$$

$$s_i = \left\{ \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [I^i(x, y) - E_i]^3 \right\}^{\frac{1}{3}}.$$

The choice of the color space where the color moments are computed depends on the application, and usually the Red, Green, Blue (RGB) color space is not the best choice. For image similarity measurement, the Hue, Saturation, Value (HSV) color space shows good performance [31], and for high-level concept classification, the LUV color space is preferred [32].

### Gabor texture

In [33], Manjunath and Ma used Gabor wavelet features for texture analysis and achieved promising results. A 2-D complex Gabor function  $g(x, y)$  is given by,

$$g(x, y) = s(x, y)w(x, y), \quad \text{where}$$

$$s(x, y) = \exp(2\pi j Wx),$$

$$w(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right].$$

$s(x, y)$  is a complex sinusoidal, known as the carrier, and  $W$  is its frequency.  $w(x, y)$  is a 2D Gaussian-shaped function known as the envelop, and  $\sigma_x$  and  $\sigma_y$  are the standard deviations in two dimensions. The Fourier transform of  $g(x, y)$ , denoted by  $G(u, v)$ , is

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\},$$

where  $\sigma_u = 1/(2\pi\sigma_x)$ , and  $\sigma_v = 1/(2\pi\sigma_y)$ . Gabor functions form a complete but non-orthogonal basis set. A class of self-similar functions, known as Gabor wavelets, can be derived from the mother Gabor wavelet  $g(x, y)$ ,

$$g_{mn}(x, y) = a^{-m} g(a^{-m}(x \cos \theta_n + y \sin \theta_n), a^{-m}(-x \sin \theta_n + y \cos \theta_n)),$$

where  $\theta_n = \frac{n\pi}{K}$ ,  $a > 1$ ,  $m, n \in \text{Integer}$ .

$K$  is the total number of orientations. It is clear that  $g_{mn}(x, y)$  is a scaled (by  $a^{-m}$ ) and rotated (by  $\theta_n$ ) version of  $g(x, y)$ . The Gabor wavelets provide a localized frequency analysis.

Texture analysis uses a subset of  $g_{mn}(x, y)$ , which covers a range of frequencies at certain scales ( $S$ ) with a certain number of orientations ( $K$ ). A strategy for designing this subset is to ensure that the half-peak magnitude support of the filter responses in the frequency domain touches each other.

The Gabor wavelet transform for an image  $I(x, y)$  is defined as follows,

$$W_{mn}(x, y) = \iint I(x', y') g_{mn}^*(x - x', y - y') dx' dy',$$

$m = 1, \dots, S; n = 1, \dots, K.$

where  $*$  denotes the complex conjugate. The mean and the standard deviation of the transform coefficients can be used to represent the texture information.

$$\mu_{mn} = \iint |W_{mn}(x, y)| dx dy, \quad \sigma_{mn} = \sqrt{\iint (|W_{mn}(x, y)| - \mu_{mn})^2 dx dy}.$$

Zavesky et al. [32] used a combination of four scales and six orientations to extract texture features for concept classification task.

#### Edge direction histogram

An edge direction histogram denotes the distribution of edge directions in an image. Figure 13.5 illustrates how this feature is extracted. The input image is shown in Figure 13.5a and b presents the edge

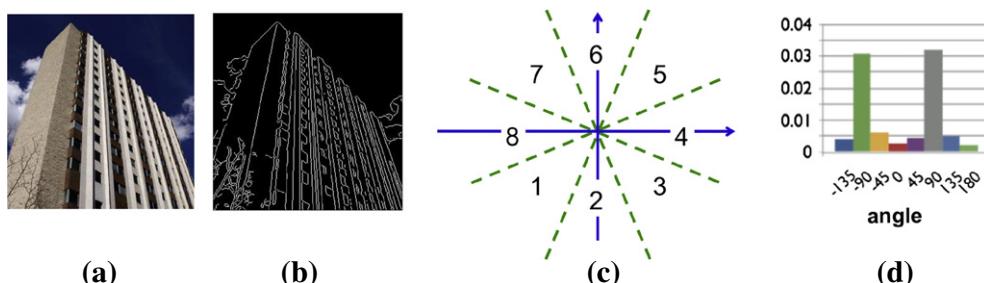


FIGURE 13.5

Computation of edge direction histogram demonstrating angular components of a building.

detection result using the Canny edge detector. Then for each edge pixel, the edge direction is computed using Sobel operators. The entire edge direction range  $[-\pi, \pi]$  is split into  $N$  bins, where  $N = 8$  in Figure 13.5c. Finally, the histogram of edge direction is counted for these  $N$  bins. Usually, the histogram is normalized by the number of pixels in the image to cancel the effects of the differing image sizes. Figure 13.5d shows the normalized histogram. In the work reported in this chapter, 16 bins are used to quantize the edge direction and one extra bin is used for non-edge points.

### Improved global visual features

Since the global visual features ignore the spatial information embedded in the image, their discriminative capability is affected. One way to cope with this shortcoming is to divide the entire image into grids (e.g.,  $4 \times 4$  grids). Global visual features are first computed from each grid, and then features from all grids are concatenated into one global visual feature vector for the image. In such way, certain spatial information is embedded in the overall visual features. Based on the global features just discussed, each grid produces 50 features (9 color moments, 24 Gabor textures, and 17 edge direction histogram), and the overall dimension of the global feature for one keyframe is 800 (assuming the number of grids is 16).

#### 5.13.2.4.2 Local visual features

Local visual features have been successfully utilized in many computer vision applications like object recognition, image stitching, pedestrian tracking, etc. A great survey on the local visual features can be found in [34]. Usually, the local feature extraction is composed of two stages: (1) Locating the keypoints and (2) Computing a descriptor for each keypoint. In certain applications, the first stage can be replaced by a dense sampling mechanism, where nodes of a regular grid over the image are treated as keypoints. One extreme case is that the DAISY feature [35] allows descriptors be calculated for every image pixel, which enables a wide-baseline stereo matching. In the following two subsections, details on these two stages are discussed.

### Keypoint detector

The first step is to locate the keypoints that can be robustly reproduced in transformed images. This section introduces two representative keypoint detectors: Harris corner detector [36] and the Scale Invariant Feature Transform (SIFT) [37]. Additional keypoint detector methods, such as Features from Accelerated Segment Test (FAST), Maximally Stable External Region (MSER), etc., can be found in [38,39].

The central idea of Harris corner detector is that the image intensity value changes in multiple directions at a corner point. It is based on the local auto correlation function of a signal, which measures the local changes of the signal with patches shifted by a small amount in different directions. The difference between the original patch and moved patch is computed as follows:

$$\begin{aligned} E(u, v) &= \sum_{x,y} w(x, y)[I(x+u, y+v) - I(x, y)]^2 \\ &\approx \sum_{x,y} w(x, y)[I(x, y) + uI_x + vI_y - I(x, y)]^2 \end{aligned}$$

$$\begin{aligned} &\approx [u \ v] \left( \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \\ &\approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix}, \end{aligned}$$

where  $(u, v)$  is the patch displacement in the vertical and horizontal directions,  $w(x, y)$  is a window centered at  $(x, y)$ , serving as a mask,  $I(x, y)$  is the image intensity at  $(x, y)$ , and  $M$  is the Harris matrix. Let's assume that the two ordered eigenvalues of  $M$  are  $\lambda_1$  and  $\lambda_2$  ( $\lambda_1 \geq \lambda_2$ ). If both  $\lambda_1$  and  $\lambda_2$  are close to zero, the pixel  $(x, y)$  is not an edge or corner point; If  $\lambda_1$  is large and  $\lambda_2$  is close to zero, the pixel  $(x, y)$  is an edge point; If both  $\lambda_1$  and  $\lambda_2$  are large, the pixel  $(x, y)$  is corner point. A measure for detecting corner is defined as  $R = \det(M) - k(\text{trace}(M))^2$ , and it is obvious that  $\det(M) = \lambda_1 \lambda_2$ , and  $\text{trace}(M) = \lambda_1 + \lambda_2$ . Based on the previous observation, when  $R$  is bigger than a threshold, the point  $(x, y)$  is detected as a corner pixel.

SIFT keypoint detector belongs to a more general method called blob detection which detects the regions in an image that differ from the surrounding regions. Specifically, SIFT keypoint detector finds the local extreme of Difference of Gaussian (DoG) values both in scale and in space. The DoG of an image  $I$  is computed as follows:

$$\begin{aligned} D(x, y, \sigma) &= L(x, y, k\sigma) - L(x, y, \sigma), \quad \text{where} \\ L(x, y, \sigma) &= G(x, y, \sigma) * I(x, y), \end{aligned}$$

$G(x, y, \sigma)$  is the 2D Gaussian filter with a variance  $\sigma$ , and  $L(x, y, \sigma)$  is the convolution of the input image  $I$  with the Gaussian filter. The DoG is a function of three variables ( $x$  and  $y$  in the spatial domain, and  $\sigma$  in the scale domain). A subpixel refinement for the local extreme can be achieved by quadratic interpolation. The dominant gradient orientation in a window around the keypoint  $(x, y, \sigma)$  is treated as its orientation. This orientation and the scale are used to normalize the final descriptor such that it is orientation and scale invariant.

### Feature descriptor

The second step is to compute a descriptor for each keypoint. Among a handful of different local visual features, the following are widely adopted: Scale Invariant Feature Transform (SIFT) [37], the Speeded Up Robust Features (SURF) [40], Gradient Location and Orientation Histogram (GLOH) [41], Binary Robust Independent Elementary Features (BRIEF) [42], Oriented BRIEF (ORB) [43], Binary Robust Invariant Scalable Keypoints (BRISK) [44], Fast Retina Keypoint (FREAK) [45], DAISY [35], PCA-SIFT [46], Local Binary Patterns (LBP) [47], etc. This section briefly introduces SIFT and FREAK features.

The SIFT descriptor is a statistic of the gradient orientations around the corresponding keypoint, where the size of the region is determined by the scale of the keypoint. First, the gradient magnitudes and orientations are computed at each sample point determined by the scale of the keypoint. Second, the magnitudes are weighted by a Gaussian window centered at the keypoint for two purposes: assign less weight to the gradients far away from the keypoint and reduce the sensitivity of the descriptors to the keypoint position. Third, the region is split into  $4 \times 4$  subregions, and for each subregion a gradient orientation histogram with 8 bins is created. Finally, these 16 sets of histograms are concatenated to create

a 128-dimension feature vector. The gradient orientations are rotated relative to the keypoint orientation in order to achieve orientation invariance in addition to scale invariance. A few other local visual features are inspired by the SIFT descriptors. For example, SURF is basically an efficient approximation of SIFT by using integral image. GLOH adopts a log-polar location grid and 16-bin gradient histogram around the keypoint, and the raw feature is mapped to 128 dimensions by PCA. In PCA-SIFT, the raw features are horizontal and vertical gradients sampled at  $39 \times 39$  locations, and then the 3,042-dimension feature is transformed to 36 dimensions by PCA.

The FREAK feature is inspired by the characteristics of retina in human visual system. It generates a cascade of binary strings by comparing the image intensities over a retinal sampling pattern, which is the main uniqueness of FREAK. Different sampling patterns are used in other local visual features, for example, BRIEF and ORB use random pairs, and DAISY and BRISK use circular patterns. In FREAK, the sampling pattern is similar to the retinal ganglion cell distribution: the spatial resolution is fine near the center and is coarser when moving away from the center. The receptive fields are arranged in a circular manner. They are overlapped and their sizes change exponentially based on its distance from the center. Few dozen of receptive fields lead to thousands of pairs for intensity comparison. Instead of using all of them, the FREAK feature learns the best 512 pairs from training data. These pairs are ordered in a coarse to fine structure, and the intensity comparisons of them constitute the 512-bit FREAK feature.

OpenCV [48] implemented many of the above-mentioned keypoint detection methods and feature descriptor extraction methods. The default parameters for the SIFT feature computation program can generate thousands of SIFT features for a single image, which brings more computational complexity in the matching step. If the edge threshold is set to be 5 and the peak threshold to be 7, this brings the number of keypoints to about 200 for a frame of size  $320 \times 240$ .

### 5.13.2.5 Visual feature indexing and search

How to index and search the reference visual features plays a key role in the overall performance of the video copy detection system. This section first introduces methods for indexing and searching features in the original feature space and then shows different ways to improve the scalability.

#### 5.13.2.5.1 Index and search in the original feature space

The  $k$ -d tree is a data structure for storing  $k$ -dimensional points, and it is one of the first attempts for searching the visual feature in the feature space. The  $k$ -d tree is a binary tree where each node is a  $k$ -dimensional point (e.g., for SIFT feature,  $k$  is 128). Every internal node implicitly divides the space into two parts with a hyperplane, which passes the point associated with the node and is perpendicular to the direction specified by the chosen dimension's axis. Points to the left of this hyperplane represent the left subtree of that node and points right of the hyperplane are represented by the right subtree. The choice of the splitting dimension can be selected either in round-robin manner or by the dimension with maximum variance among the current subcell. For a balanced  $k$ -d tree, the point with the median value in the selected dimension is inserted to the node. This process iterates for the left and the right subtree until the leaf node is reached, where only one point exists in the current subcell.

Searching the nearest neighbors of a query point in a  $k$ -d tree begins with descending the query point in the tree and finding the cell that contains the query point. The points contained in this cell usually are not the nearest neighbors of the query point. Instead, a list of candidate nodes in the tree

are dynamically generated based on their distances to the query point while it passes through the  $k$ -d tree, and these nodes are checked to pinpoint the nearest neighbors. The complexity of the search is somewhere between  $\log(N)$  and  $N$ , where  $N$  is the total number of reference points.

$k$ -d tree method is efficient in low dimensions, but the performance drops in high dimension scenarios. Instead of finding the exact nearest neighbors, locating the approximate nearest neighbors greatly increases the efficiency with a tolerable loss in accuracy [49]. Multiple randomized  $k$ -d trees may also be used, where the split dimension at each internal node is randomly selected from the top D dimensions with the greatest variances. Then a shared priority queue is maintained while searching all trees. The approximation is achieved by searching only a fixed number of leaf nodes. Experiments show that it is faster than linear search by several orders of magnitude yet still achieves good performance.

### 5.13.2.5.2 Locality sensitive hashing (LSH)

Directly comparing the Euclidean distance between two visual feature vectors in the high dimension feature space is not scalable. This system utilizes Locality sensitive hashing (LSH) [50] for efficient visual feature matching.

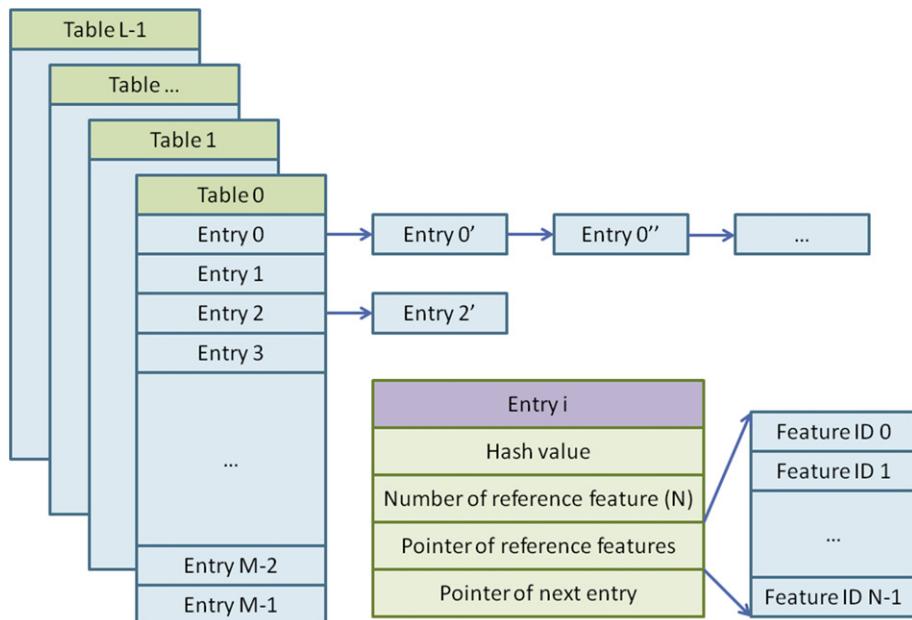
The idea of the Locality sensitive hashing is to approximate the nearest-neighbor search in high dimensional space. The hash function possesses the desirable property that when two vectors are closer in the feature space, their hash values are more likely to be the same, and when they are farther away in the original vector space, their hash values are less likely to be the same. Using a Hash function, the complex high dimension vector distance comparison is converted into one integer comparison, which is very efficient. Each hash function  $h_{\mathbf{a},b}(\mathbf{v})$  maps a vector  $\mathbf{v}$  onto the set of integers (bins),

$$h_{\mathbf{a},b}(\mathbf{v}) = \left[ \frac{\mathbf{a} \cdot \mathbf{v} + b}{w} \right],$$

where  $w$  is a preset bucket size,  $b$  is chosen uniformly in the range of  $[0, w]$ , and  $\mathbf{a}$  is a random vector following a Gaussian distribution with zero mean and unit variance.

Two additional parameters to tune the hashing performance are: (1) Combine  $k$  parallel hashing values to increase the probabilities that vectors far away fall into different bins; (2) Form  $L$  independent  $k$ -wise Hash values to reduce the probability that similar vectors are unluckily projected into different bins. In this work, for the SIFT/SURF features, the following parameters give satisfactory results:  $w = 700$ ,  $k = 24$ , and  $L = 32$ . This basically creates 32 independent Hash values for each of the local visual features. For the global visual features,  $w = 800$ ,  $k = 24$ , and  $L = 64$  are used.

A standard hashing approach (see Figure 13.6) can dramatically improve the efficiency of LSH indexing and query. A classic hash table indexes computed LSH values;  $L$  hash tables index the  $L$  sets of LSH values. The size of hash table ( $M$ ) depends on the number of unique LSH values, and the tolerance of hash value conflicts. The original LSH value is mapped to the entry in the table by a hashing function (32-bit integer to  $[0, M]$  mapping), and conflicting entries are linked through pointers (e.g., Entry 1' and Entry 1''). The detailed data structure of each entry is shown on the right-hand side of Figure 13.6. The first field keeps the original LSH value, the second field counts the number of reference local visual features that are mapped to this entry, the third field saves the list of these reference feature IDs, and the last field is a pointer to the next entry, in case there is a conflict. Normally, the last field is set to NULL.

**FIGURE 13.6**

LSH hash indexing.

While indexing all LSH values in the reference dataset, the  $L$  hash tables are populated, and the arrays of reference local visual feature IDs in each entry are sorted based on their video, frame, and keypoint IDs. Compared to the binary search method, this implementation maintains a near constant time query complexity, and it increases the LSH query speed significantly.

Once hash tables are computed, the tables can be trimmed based on the number of reference keypoints. If this number is too high, it means the corresponding local visual feature is not descriptive, and it can be removed from the table. The trimming process benefits the overall system in two ways: (1) increases the query speed and (2) improves the robustness of local visual feature based query.

While this indexing method gives highly accurate performance, the scalability is still limited compared to other approaches. The next section introduces the bag of visual words (BoW) method, which has been widely adopted in recent years.

### 5.13.2.5.3 Bag of visual words

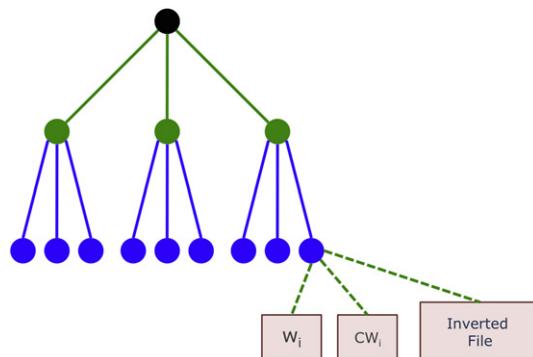
In [51], a new method is proposed to treat images as documents as in the classic information retrieval task, and treat the local visual features as visual words. The image is then represented by the distribution or histogram of the embedded words. With this mechanism, the mature techniques, including the term frequency inverse document frequency (TFIDF), the stop word removal, the N-grams, etc., can be easily applied to large-scale image retrieval. Recent progresses in this area were summarized in two review papers [52, 53]. The key challenges are how to generate the visual-word vocabulary and how to quantize

visual features fast. Two popular methods for the BoW (bag of words) approach are  $k$ -means clustering and vocabulary tree [54].

The basic way to generate a visual-word vocabulary is by  $k$ -means clustering. The  $k$  codewords are randomly selected initially, and in each iteration, every training visual feature sample is assigned to its nearest codeword, and then the codeword is replaced by the mean of all visual feature samples assigned to it. This iterative process terminates when all  $k$  codewords converge. The goal is to minimize the total distance of all samples to their assigned codewords. The results of  $k$ -means depend on the choice of the original set of codewords, and the procedure only guarantees a locally optimum solution. In real applications, several  $k$ -means clustering operations are performed using different sets of initial codewords, and the one with the minimum total distance is selected.  $k$ -means is a straightforward method, but it is not scalable when  $k$  is large (e.g., in the order of millions). The reason is that the feature quantization step needs to measure the distance between the input visual feature and all  $k$  codewords, and this is slow especially when the feature dimension is also high.

One possible solution to address the scalability issue of  $k$ -means clustering is the vocabulary tree method proposed in [54]. Vocabulary tree is based on hierarchical  $k$ -means. The tree structure is defined by two parameters: the tree branching factor  $B$  (the maximum number of children for each internal node), and the depth of the tree  $D$  (the maximum number of edges between a leaf node and the root node). The maximum number of leaf nodes is  $B^D$ . A vocabulary tree with a branching factor of 3 and a depth of 2 is shown in Figure 13.7. First, all training visual features are assigned to the root node. A  $B$ -means clustering is applied on them to determine the  $B$  codewords. These  $B$  codewords are assigned to the  $B$  children nodes of the root node, and the training features belonging to each codeword are assigned to the corresponding children node as well. This  $B$ -means clustering procedure is applied to each child node recursively until the maximum tree depth ( $D$ ) is reached or other conditions are met, for example, the training features associated with the current node are less than a preset threshold (e.g., 16). After this hierarchical  $k$ -means clustering, each node ( $i$ ) has a codeword  $CW_i$ .

Once the tree structure and the codeword for each node are finalized, it is possible to index the reference keyframes. Assume there are  $N$  reference keyframes, each denoted by  $F_j$ ,  $j = 1, \dots, N$ .



**FIGURE 13.7**

Structure of vocabulary tree, shown here with a depth of 2 and branching factor of 3.

Every local visual feature of  $F_j$  is quantized to one codeword (when only leaf node is considered) or a set of codewords (when both leaf nodes and internal nodes are considered) based on how the vocabulary tree is traversed. Each node maintains an inverted file list to record all reference keyframes whose local features pass the node in the quantization process. In implementation, only leaf nodes need to maintain this inverted file list, since internal nodes can virtually aggregate the lists of their descendant nodes. After indexing all reference images, the weight ( $w_i$ ) of each node ( $n_i$ ) can be determined by the traditional inverse document frequent method:  $w_i = \log(N_i/N)$ , where  $N_i$  is the size of the associated inverted file list. The node weight is used to further refine the choice of visual words in the system. For example, setting all weights that are less than a preset threshold to zero will effectively remove the noisy visual words that appear in too many keyframes. Setting the weights of certain internal nodes to zero will control the number of visual words generated for one visual feature. Then, the number of visual words (nodes with a non-zero weight) is finalized, which is denoted by  $M$  (the vocabulary size).

To make searching more efficient, it is necessary to compute the reference vectors in advance. For a reference keyframe  $F_j$ , the dimension of its feature vector  $\mathbf{r}_j$  is  $M$ , and each entry  $r_j^i$  is the product of  $n_j^i$  and  $w_i$ , where  $n_j^i$  is the frequency of word  $n_i$  that appears in  $F_j$ . In most applications, most entries in vector  $\mathbf{r}_j$  are zero, and the vector is very sparse.

For searching the similar reference keyframes for a query keyframe  $Q$ , the first step is to determine its vector representation  $\mathbf{q}$ , and then among all reference keyframes collected from the inverted file list while quantizing each visual feature of  $Q$ , return those with lowest distance from  $Q$ . Similar to the reference keyframe case,  $\mathbf{q}$  is a vector with entries  $q^i$ , which is the product of  $w_i$  and  $m_q^i$ , the frequency of word  $n_i$  appears in  $Q$ . The distance between  $\mathbf{q}$  and  $\mathbf{r}_j$  can be computed as follows,

$$d(\mathbf{q}, \mathbf{r}_j) = \left\| \frac{\mathbf{q}}{\|\mathbf{q}\|} - \frac{\mathbf{r}_j}{\|\mathbf{r}_j\|} \right\|,$$

where the norm is determined based on the application (e.g.,  $L_2$ -norm).

To further boost the query speed, there are two implementation considerations. First, the inverted file list needs to be sorted based on the reference keyframe ID such that merging of multiple inverted file lists can be efficiently achieved. Second, the feature vectors are sparse, and the distance between two sparse vectors can be computed quickly by sparse representation, where only word id and word weights are stored for non-zero entries.

### 5.13.2.6 Local visual feature match verification based on geometric constraint

Keyframe matching based on LSH Hash or BoW method is efficient, yet not reliable enough. There are two issues: (1) the original visual feature matching by Euclidean distance is not reliable, especially when the number of local visual features is large and various transformations may introduce noise, (2) it is possible that two visual features that are dissimilar are mapped to the same Hash value or codeword. Therefore, an additional mechanism is utilized to validate the list of retrieved reference keyframes produced in the last section. In this work, RANdom Sample Consensus (RANSAC) [55] is utilized for this purpose.

RANSAC is an iterative method for estimating model parameters from observed data with outliers. Here, RANSAC is used for estimating the affine transform that maps the keypoints in the original reference keyframe to those in the query keyframe. The affine transform is able to model the geometric

changes introduced by following transforms: PiP, shift, aspect ratio change, etc. Specifically, the keypoint at pixel  $(x, y)$  in the reference keyframe is mapped to pixel  $(x', y')$  in the query keyframe by the following formula,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

To determine the affine model parameters  $(a, b, c, d, e, f)$ , three pairs of keypoints, where the keypoints in the reference keyframe are not co-linear, are required. For a pair of keyframes, the detailed RANSAC procedure is as follows,

1. Randomly select three pairs of matching keypoints.
2. Determine the affine model.
3. Transform all keypoints in the reference keyframe into the query keyframe.
4. Count the number of keypoints in the reference keyframe whose transformed coordinates are close to the coordinates of their matching keypoints in the query keyframe. These keypoints are called inliers.
5. Repeat steps 1–4 for a certain number of times, and output the maximum number of inliers.

Figure 13.8 presents an example of RANSAC verification where the adopted local visual feature is SIFT. Figure 13.8(a) displays a query keyframe (on the left side) and a reference keyframe side by side, and links all original matching keypoints by lines. Figure 13.8(b) shows the matching keypoints after RANSAC verification. All of the wrong matching keypoints are reliably removed.

RANSAC is robust, but not very efficient. To overcome this problem, a weak verification method can be used with less effectiveness yet much faster. The idea is to examine the histogram of scale ratio and orientation difference of matching keypoints, and significant peaks indicate a strong geometric congruence. Readers may refer to [56] for more details.

### 5.13.2.7 Frame level result fusion

To cope with the visual effects introduced by various transformations, both the reference keyframes and the query keyframes are normalized. The choice of combinations of reference/query normalization is driven by the application. For near-duplicate detection, this step may not be required; while for more



**FIGURE 13.8**

Feature matching verification by RANSAC. (a) Original feature matching result. (b) Feature matching result with RANSAC.

**Table 13.1** Normalization Pairs of Query and Reference Keyframes

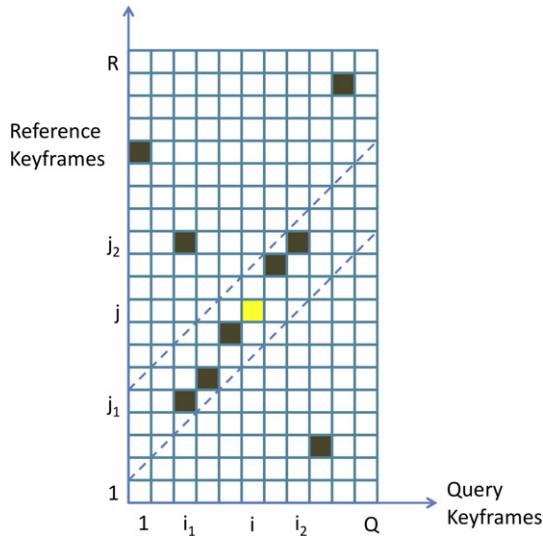
Pair	Query keyframes	Reference keyframes
1	Original	Original
2	Flipped	Original
3	Original	Encoded
4	Flipped	Encoded
5	Picture in picture (PiP)	Half
6	PiP and flipped	Half
7	Letterbox removed	Original
8	Letterbox removed and flipped	Original
9	Intensity equalized	Original
10	Intensity equalized and flipped	Original
11	Blurred	Original
12	Blurred and flipped	Original

difficult cases, like the TRECVID CCD evaluation task, 12 combinations are evaluated in this work (see [Table 13.1](#)). The merging process is to simply combine these 12 lists into 1 list. If one reference keyframe appears more than once in the 12 lists, its new relevance score is set to be the maximum of its original relevance scores, otherwise, its relevance score is the same as the original one.

### 5.13.2.8 Video level result determination

Based on the results of all keyframes in a query video, a list of best matching videos is determined. Consider one query video and one reference video, where the query video has  $Q$  keyframes, and the reference video has  $R$  keyframes, the relevance score between query keyframe  $i$  and reference keyframe  $j$  is denoted by  $S(i, j)$ . The timestamp of query keyframe  $i$  is  $QT(i)$ , and that of reference keyframe  $j$  is  $RT(j)$ . [Figure 13.9](#) illustrates an example of the relevance score matrix, where all non-zero entries are marked by gray squares.

For a matching keyframe pair  $(i, j)$ , which is highlighted in [Figure 13.9](#), the timestamp difference is first computed:  $\Delta = RT(j) - QT(i)$ . Then all matching keyframes whose timestamp difference is in the range of  $[\Delta - \delta, \Delta + \delta]$  are found, where  $\delta$  is set to 5 s in this work. In [Figure 13.9](#), the time difference range is marked by two dashed lines. The extended relevance score for pair  $(i, j)$  is simply the summation of the relevance scores of these filtered pairs. To reduce the impact of the number of matching query keyframes,  $N$ , the extended relevance score is normalized by  $1 / \log(N + 1)$ . The video level matching segments determined by pair  $(i, j)$  are query keyframes  $[i_1, i_2]$  and reference keyframes  $[j_1, j_2]$ , and  $N = i_2 - i_1 + 1$ . Once the extended relevance scores for all matching pairs are computed, the pair with the maximum score is picked as the video level matching between the corresponding reference video and the query video.

**FIGURE 13.9**

Video level result fusion.

### 5.13.2.9 Score normalization for visual-based results

Assuming a query video has  $N$  matches, and its original visual matching scores are  $s'_v(i)$ ,  $i = 0, \dots, N - 1$ , the normalized scores  $s_v(i)$  are computed by the formula

$$s_v(i) = \alpha(i) \times \text{sigmoid}(s'_v(i)),$$

$$\text{where } \alpha(i) = \begin{cases} \frac{M - i}{M} \times \frac{s'_v(i)}{\sum_{j=0}^{M-1} s'_v(j)}, & \text{for } 0 \leq i \leq M, \\ \alpha(M - 1), & \text{for } i \geq M, \end{cases}$$

where  $M$  is a preset number, empirically 8 in this system. When  $N$  is smaller than  $M$ , the last match score is repeated, and the list of match videos is extended to  $M$ . As reflected by the formula, the weight  $\alpha(i)$  is determined by both the rank,  $i$ , and the match score  $s'_v(i)$ . Score normalization is inspired by the method reported in [57], but differing from the prior implementation, the sigmoid function of the original match score is used. The range of normalized scores is  $[0, 1]$ .

---

### 5.13.3 Audio-based video copy detection

While visual-based copy detection is usually slow given the huge amount of data the system needs to process, it handles cases where the audio has been replaced or where there is no audio track associated with video. On the other hand, analyzing the audio information can provide valuable information for copy detection. The main advantage of utilizing audio information for video copy detection is the efficiency.

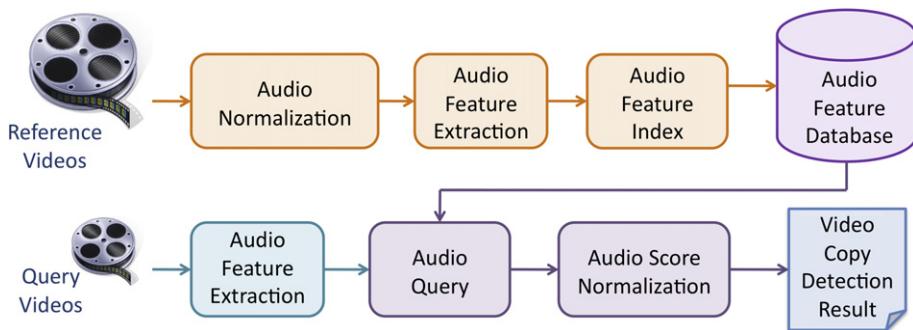
**FIGURE 13.10**

Diagram of the audio-based video copy detection algorithm.

### 5.13.3.1 Audio copy detection module

Figure 13.10 shows a general block diagram of audio-based video copy detection system. Similar to the visual-based approach, normalizing reference audio enables the system to cope with various audio transformations that query videos have gone through. For example, audio compression and bandpass filtering, etc. The choice of normalization method depends on the applications. Audio features are extracted from normalized reference audio and then the features are indexed and stored in the audio feature database. For query videos, audio features are extracted first and then they are used to query the reference feature database. For each normalization type of the reference audio, a list of matching audio is created. Then these matching scores are fused and normalized. The audio matching scores can be normalized in the same way described in the visual-based approach. Finally, the video copy detection results are generated.

### 5.13.3.2 Audio normalization

In real applications, there are many audio transformations introduced in the audio editing and producing procedure, including audio compression, re-sampling, bandpass filtering, companding, speech insertion, background music addition, etc. In this work, two audio normalizations are considered for reference audio: (1) Compression, where audio is compressed in MP3 format with a low bit rate of 16 Kbps; (2) Bandpass filtering, where the bandpass filter covers the frequencies between 500 Hz and 3000 Hz.

### 5.13.3.3 Audio features

In this section, both general-purpose acoustic features and audio fingerprint for audio content identification are introduced.

#### 5.13.3.3.1 Acoustic features

Mel-frequency cepstral coefficients (MFCC) or cepstral coefficients (CC) [9] are widely used for speech recognition and speaker recognition applications. While both of these provide a smoothed representation of the original spectrum of an audio signal, MFCC further considers the non-linear property of the human

hearing system with respect to different frequencies. Given the success of MFCC in the speech domain, it is also very popular in other audio signal processing, for example, audio/music classification, audio content segmentation, speaker segmentation, language identification, etc. For audio-based video copy detection, MFCC, equalized MFCC (the mean of cepstral features in each audio is subtracted to generate zero mean features), and Gaussianized MFCC (non-linear transformation is applied to the cepstrum such that the features have a Gaussian distribution) were adopted in [19]. MFCC has 13 dimensions, and when the first order temporal derivative of MFCC is considered, the total feature dimension becomes 26. The difference between any two MFCC feature vectors can be measured by their Euclidean distance.

Other acoustic features that can be used for audio content identification include Fourier coefficients, Linear Predictive Coding (LPC) coefficients, Modulated Complex Lapped Transform (MCLT) [58], etc.

#### **5.13.3.2 *Audio fingerprint***

The general framework for most audio fingerprint extraction schemes is as follows [59]. Audio is first segmented into frames, and acoustic features are extracted for each frame. Then, these features are mapped into an audio fingerprint that is a compact representation. Three audio fingerprint extraction methods are introduced below and an overview of many more audio fingerprinting can be found in [13].

In [57], a 15-bit fingerprint is extracted for every audio frame, which is 25 ms long with 15 ms overlap between adjacent frames (the frame rate is 100 frames per second). The audio signal first goes through a low pass filter with a cut-off frequency of 4 KHz. Similar to traditional speech signal processing, a pre-emphasis by a first order high pass filter is applied to the audio input. The spectrum between 300 Hz and 3000 Hz is split into 16 subbands in Mel-scale. Then the energy differences between subbands are used as the fingerprint. In [22], a similar audio fingerprint is devised, where the main differences are as follows. (1) The fingerprint has 16 bits, computed from 17 subbands covering 300–6400 Hz in bark-scale. Since the fingerprint covers a wide spectrum, its discriminative capability may be improved. (2) Each frame is 32 ms long. While the audio fingerprint is very easy to compute and compare, it only contains 16 bits (64K different values). For a collection of hundreds of hours of audio (tens of millions of frames), and on average, thousands of frames share the same fingerprint. It is obvious that a single fingerprint is not sufficient to identify an audio clip, and the audio-based search has to rely on the temporal constraints – the sequence of audio fingerprints, as discussed in the following section.

Baluja and Covell proposed an audio fingerprinting named Waveprint for audio identification in [60]. The novelty of this scheme is that it relies on computer vision techniques to generate compact fingerprints from the audio spectrogram, which is treated as an image. Specifically, a spectrogram is computed for the input audio, and it is decomposed using the multi-resolution Haar wavelets. To reduce the effect of noise, only the top Haar wavelets according to their magnitudes are kept. Instead of keeping the actual values of the wavelet coefficients, their signs are used to save memory usage. Then Min Hash technique is applied on this binary vector multiple times ( $p$  times) to create a set of  $p$  integers as the audio fingerprint. Basically, the Min Hash method permutes the binary vector positions in a pseudorandom order, and measures the first position that 1 occurs. The  $p$  dimension fingerprint can be further compressed with the Locality Sensitive Hashing (LSH) to reduce the number of comparisons in retrieval stage.

An interesting local audio fingerprint, named MASK (Masked Audio Spectral Keypoint), was proposed in [61]. The MASK fingerprint extraction method is composed of four steps. First, the audio signal is transformed from the time domain to the frequency domain by the short-time Fourier transform (STFT), and then converted into 18 bands in Mel-scale. Second, spectral salient points are selected, which are the local spectral peaks in time and/or frequency domains. Third, for each salient point, a

mask that covers 5 Mel bands and 19 frames is applied around it and a number of regions with different sizes and locations within the mask are determined. Some of the regions have overlap with each other. Finally, the average energy values of all regions are computed and a fixed length binary descriptor is constructed by comparing the energy of a selected pair of regions. This local binary descriptor is used as audio fingerprint for indexing and retrieval.

Shazam, one of the pioneering music search mobile apps, relies on a more descriptive audio fingerprint [11]. There are three basic steps: (1) The spectrogram is computed for the input audio by short-time Fourier transform. It represents the audio energy at certain (time, frequency) coordinate. (2) Peaks, the time-frequency points whose energy is the local maximum in a region centered around the point, are selected to construct a constellation map. (3) For an anchor point in the constellation map, a target zone is constructed in the constellation map based on its time and frequency. The anchor point and any point in the target zone form a pair, and their frequencies and time difference are used to create a 32-bit unsigned integer (fingerprint). Wang [11] showed that these fingerprints are quite reproducible, even in the presence of noise and compression. Such a method is very effective for music and songs, but it is not clear whether it still performs well for other types of audio, for example, speech, where the peaks in spectrogram may not be dominant.

#### 5.13.3.4 Audio index and search

This section demonstrates search with the 16-bit fingerprint computed in the previous section. The same hash indexing method previously described is adopted for audio, where only one table ( $L = 1$ ) with a fixed size of 64K (due to a 16-bit fingerprint value) is utilized. Here, the challenge is to efficiently implement the audio fingerprint query, where matches must have temporal consistency. The slow, brute force approach uses a sliding window and counts the audio frames matching within this window. This

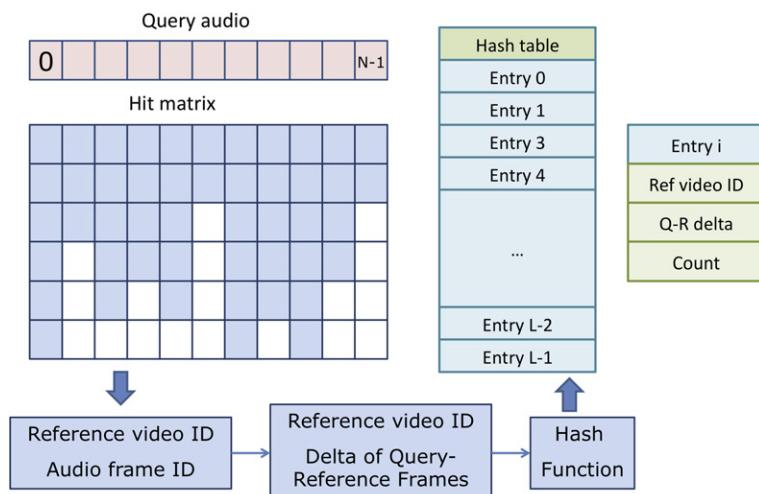


FIGURE 13.11

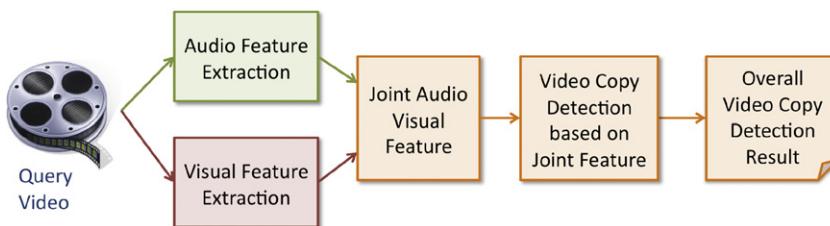
Audio feature query by fingerprint.

system utilizes query frame results for a faster hashing method, shown in Figure 13.11. A query audio is shown at the top left corner, assuming there are  $N$  frames in this query. Hits for each query frame can be easily retrieved from the indexing method shown in Figure 13.11. The results are plotted in the Hit matrix under the query audio. Each gray block represents a hit, which is specified by a reference video ID and a reference audio frame number. Then for each hit, the difference of the query audio frame and the reference audio frame (so called Q-R delta) is computed. The concatenation of the reference video ID and the difference between the query and reference frame numbers form a key to populate a new counting hash table. Each hash entry represents the frequency of hits for a certain reference video with a certain temporal offset and entries with the greatest values are the detection results. Using this technique, one scan of the frame-based matches achieves the same results as the sliding window approach. Finally, the matching scores are normalized in the same way described in visual approach.

## 5.13.4 Joint audio- and visual-based video copy detection

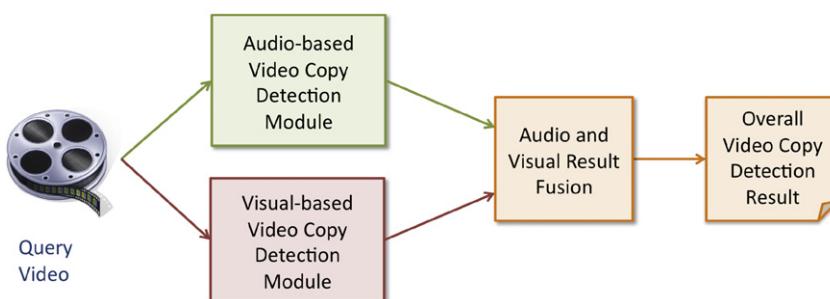
### 5.13.4.1 Audio and visual fusion schemes

Generally speaking, there are two main categories of schemes that fuse audio and visual information together for video copy detection task. They are early fusion and late fusion schemes [62], shown in Figures 13.12 and 13.13, respectively. The early fusion scheme combines the audio and visual feature



**FIGURE 13.12**

Early audio-visual fusion approach.



**FIGURE 13.13**

Late audio-visual fusion approach.

spaces into one joint space, and the video copy detection engine works on this joint feature space directly. This scheme is also referred to as the feature-level fusion scheme. On the contrary, in the late fusion scheme, the audio- and visual-based subsystems work independently, and the detection results from both modules are combined at the end. This scheme is referred to as the decision level fusion. [62] presented interesting observation on the choice of these two schemes for semantic concept detection applications. A late fusion scheme gives better performance for most concepts with an increased learning effort, and the improvement is more significant for those concepts that an early fusion performs better. Depending on the applications, a hybrid approach may be adopted to take advantage of both schemes.

In [18], a team from Peking University presented a video copy detection system that achieved the best overall accuracy and excellent localization precision in the TRECVID 2010 and 2011 content-based copy detection (CBCD) evaluation task. Complementary audio-visual features are exploited to construct several detectors and they are organized in a principled structure. The local visual feature based detector relies on a dense color version of SIFT descriptor. LAB color space is used and the dimension of the color SIFT descriptor is 216 since  $3 \times 3$  subpatches are considered for each keypoint and an 8-bin orientation histogram is generated for each subpatch and each color component. Bag-of-word method with a vocabulary size of 800 is adopted to efficiently index and search for these high dimensional features. The global visual feature based detector utilizes the relationship between the discrete cosine transform (DCT) coefficients of adjacent blocks. Energy in the lower frequency subbands of each block is compared to that of the neighboring block, and a 256-bit feature is determined by the comparison results. Locality sensitive hashing (LSH) is adopted to speed up the matching. The audio feature based detector employs the weighted audio spectrum flatness (WASF) feature proposed in [63]. To achieve high efficiency, these three detectors are arranged in a cascade architecture, where the most efficient audio-based detector is checked first and then the global visual feature based detector and the slowest local visual feature based detector. For a query video, once matching reference videos are found, the query process terminates, and the rest of the time-consuming detectors are skipped. [18] also proposed an efficient temporal pyramid matching scheme that partitions videos into increasingly finer temporal segments and computes video similarities over each granularity.

INRIA team is one of the pioneers in the video copy detection area and has been among the top performers in the TRECVID content-based copy detection evaluation task. In [20], a joint audio- and visual-based video copy detection system is reported. The adopted audio features are mainly the log-energies of overlapping bandpass filters. In total there are 40 subbands in the range of 500–3000 Hz on a Mel-scale. For visual-based module, the local binary pattern (LBP) features are used and they are quantized to visual word with a vocabulary size of 200,000. For further improvement of the efficiency, a 48-bit signature is computed for each point [64]. Both early and late fusion schemes are studied, and the interesting observation is that the early fusion scheme performs better than the late fusion scheme.

Liu et al. [22] adopted a very simple yet effective late audio-visual fusion approach. For each query video, the audio- and visual-based copy detection modules report a list of matches defined by a query segment, reference segment, and matching score. During merging, if query and reference segments of an audio match overlap those of a visual match and the overlapped regions are more than half of the original segment duration, then the matches are merged and the segment is inserted into the fused list. Otherwise, the original audio (or visual) match is inserted into the fused match list with its score weighted by a factor  $w_a$  (or  $w_v$ ), allowing a bias favoring either audio or visual results. Fused list query/reference segments are the temporal union of the two original query/reference segments and each fused score is a weighted sum of original scores:  $w_a$  for audio matches and  $w_v$  for visual matches. The

fused match list is sorted by the new scores, and normalized in the same way described in the visual copy detection module.

### 5.13.4.2 Multi-query result normalization and fusion

Multiple query search occurs when multiple examples or instances of an image or video to be found are available. Multi-query search often requires a secondary normalization that dynamically adjusts result scores according to their distribution. After this normalization, weighted fusion (from the previous section) or average fusion can be used to combine multiple queries.

Generally, results from copy detection algorithms can be described as heterogeneous or homogenous. Homogenous results have strong underlying local feature support and are visually similar during score-based traversal of the result list. Heterogeneous results have weak local feature support and similarity to the query often degrades quickly as results are traversed. This system uses three transformations to accommodate both distributions, empirically determined over training data. First, scores are min-max normalized. Let  $s(i)$  denote the score at rank  $i$ , then the normalized score is

$$s_r(i) = \frac{s(i) - s_{min}}{s_{max} - s_{min}},$$

where  $s_{min}$  and  $s_{max}$  are minimum and maximum list scores. Next, mean ( $\bar{s}_r$ ) and median ( $\tilde{s}_r$ ) scores are computed from the top  $M$  (empirically  $M = 10$ ) normalized scores and the variable  $\alpha$  from the formula

$$\alpha = \begin{cases} 1 - \tilde{s}_r & \text{if } \bar{s}_r < \tilde{s}_r, \\ \tilde{s}_r - 2 & \text{otherwise,} \end{cases}$$

indicates distribution shape. Finally, a transformed score

$$s_t(i) = \frac{\alpha \cdot s_r(i)}{\alpha - s_r(i) + 1}$$

is computed with shape indicator  $\alpha$  and normalized scores  $s_r(i)$  such that concave exponentials emphasize minute changes in homogenous sets and convex exponentials dampen differences in heterogeneous sets.

---

## 5.13.5 Copy and near-duplicate detection

An evaluation of the video copy detection system and its underlying algorithms for analysis, indexing, and retrieval are critical to demonstrating its effectiveness for large-scale content search. The TRECVID CCD task and a consumer photo management prototype serve this purpose and demonstrate the effectiveness of the proposed system. The following sections describe the systems built by AT&T Labs, their performance, and a working prototype using the copy detection algorithm.

### 5.13.5.1 TRECVID shot boundary detection (SBD) results

#### 5.13.5.1.1 TRECVID SBD evaluation criteria

The TRECVID SBD evaluation task measures the detection and accuracy of the shot transitions separately. The detection is measured by precision and recall. The submitted transitions are considered

correct only if they have at least one frame overlap with the reference transitions. Precision is the ratio of the number of correctly detected boundaries to the number of all detected boundaries, and the recall is the ratio of the number of correctly detected boundaries to the number of reference boundaries. F-measure is the harmonic mean of precision and recall. The detection performance is measured in two categories of shot boundaries: cuts (transition lasts five frames or less) and gradual (longer transitions). The accuracy for gradual transition detection is frame-based precision and recall. Assume that the number of frames in the detected reference transition is  $R$ , the number of frames in the detected submitted transition is  $S$ , and the number of shared frames by detected submitted transition and reference transition is  $Q$ , then the frame-based recall is  $Q/S$ , and the recall is  $Q/R$ .

#### 5.13.5.1.2 TRECVID SBD evaluation results

This section shows the results of the TRECVID 2007 SBD evaluation task. The testing dataset contains 17 sequences, totaling about 7 hour videos in both color and black/white. Participants are allowed to submit 10 runs, and Table 13.2 shows the best performance (marked in bold font) submitted for the proposed system. They are shown in four categories: the overall detection performance, the detection performance of cut and gradual shot boundaries, respectively, and the frame-based accuracy. In all categories of evaluation, this system is one of the top performers.

#### 5.13.5.2 TRECVID content-based copy detection (CCD) results

##### 5.13.5.2.1 TRECVID CCD task and evaluation criteria

In the TRECVID CCD task, query videos are generated from non-reference and/or reference videos with different kinds of transformations. TRECVID CCD considered the following 10 categories of visual transformations (TV<sup>\*</sup>) and 7 categories of audio transformations (TA<sup>\*</sup>):

- TV1: Simulated camcording.
- TV2: Picture in Picture (PiP).
- TV3: Insertions of pattern.
- TV4: Strong re-encoding.

**Table 13.2** Best Runs of the Proposed SBD System Submissions

Run	Category	Performance (%)		
		Recall	Precision	F-Measure
5	<b>Overall</b>	<b>95.6</b>	<b>95.4</b>	<b>95.5</b>
	Cut	97.9	96.6	97.2
	<b>Gradual</b>	<b>70.9</b>	<b>80.2</b>	<b>75.3</b>
	Frame based	71.8	93.3	81.2
3	Overall	95.5	95.3	95.4
	<b>Cut</b>	<b>97.7</b>	<b>96.8</b>	<b>97.2</b>
	Gradual	70.4	78	74
	<b>Frame based</b>	<b>74.2</b>	<b>93.3</b>	<b>82.7</b>

- TV5: Change of gamma.
- TV6: Decrease in quality: a mixture of 3 transformations among blur, gamma, frame dropping, contrast, compression, ratio, white noise.
- TV7: Same as TV6, but with a mixture of 5 transformations.
- TV8: Post production, including a mixture of 3 transformations among crop, shift, contrast, text insertion, vertical mirroring, insertion of pattern, and picture in picture.
- TV9: Same as TV8, but with a mixture of 5 transformations.
- TV10: Combinations of 3 transformations chosen from TV1–TV8.
- TA1: No audio transformation (nothing).
- TA2: MP3 compression.
- TA3: MP3 compression and multiband companding.
- TA4: Bandwidth limit and single-band companding.
- TA5: Mix with speech.
- TA6: Mix with speech, then multiband companding.
- TA7: Bandpass filter, mix with speech, and compression.

For TRECVID 2011 CCD, all of these audio and visual transformations except TV7 and TV9 are considered. The system normalizes query keyframes for detected transformations and transforms reference keyframes to eliminate the query transformation effects. As shown in [Table 13.1](#), the proposed system considers 12 combinations of normalized query keyframes and transformed reference keyframes. The CCD task evaluates results using two profiles: no false alarm (NoFA) and balanced; more details about each are in the next section. While generating runs, the NoFA profile contains only the best matches whose scores are higher than a certain threshold, whereas the balanced profile contains all matches that are higher than a lower threshold.

There are three performance measures specified for the CCD task: minimal normalized detection cost rate (NDCR), copy location accuracy, and processing time. NDCR is defined as

$$\text{NDCR} = P_{\text{Miss}} + \beta \cdot R_{\text{FA}},$$

where  $P_{\text{Miss}}$  is miss probability,  $R_{\text{FA}}$  is false alarm rate, and  $\beta$  is a profile parameter to control the tolerance on false alarms. Results of individual transformations within each run are evaluated separately. Different decision thresholds are applied to generate a list of pairs of increasing  $P_{\text{Miss}}$  and decreasing  $R_{\text{FA}}$ . Minimal NDCR is found for each transformation and actual NDCR is computed from the optimal threshold reported. For NoFA and balanced profiles, the parameter  $\beta$  is set to 2000 and 2, respectively.

Copy location accuracy is defined to assess the accuracy of finding the exact extent of the copy in the reference video. This is only measured for the correctly detected copies. Mean F1 (harmonic mean) score based on the precision and recall of the detected copy location in terms of frames relative to the true video segment is adopted. Efficiency is also an important criterion for evaluating an algorithm's effectiveness. Copy detection processing time is the mean time to process a query. It includes all processing from reading in the query video to the output of results.

### **5.13.5.2.2 TRECVID CCD evaluation results**

The TRECVID 2011 CCD dataset contains about 12K audio+video query videos, and 12K reference videos. In total, the system extracted  $\sim 82\text{M}$  SIFT features and  $\sim 110\text{M}$  audio features for the reference video set, and  $\sim 33\text{M}$  SIFT features and  $\sim 63\text{M}$  audio features for the query video set.

Four runs were evaluated: T1 and T3 (NoFA profile), and T2 and T4 (balanced profile). In runs T1 and T2, only seven combinations of video transformations (pairs 1, 2, 3, 8, 9, 10, and 11 listed in [Table 13.1](#)) and two combinations of audio transformations (original query vs. original reference, and original query vs. compressed reference) were used [22]. The choice of these combinations was determined by the performance of TRECVID 2010 dataset. For the remaining two runs, all combinations of video and audio transformations were used. For runs in the NoFA profiles, the audio score weight  $w_a$  was 1.45 and the video weight  $w_v$  was 0.55, and for runs in the balanced profile, the weights were 1.4 and 0.6 accordingly. These parameters are also determined by TRECVID 2010 dataset.

### Effects of multimodal fusion

The Hash indexing trimming threshold  $T_N$  impacts the query speed significantly. In this work,  $T_N$  was 511, which increases the query speed by three times and improves the NDCR by 4%. Further investigation for optimal NDCR performance of audio-based and visual-based approaches is summarized in [Table 13.3](#). From this table, it is clear that fusing detection results of more pairs of transformations (audio/video) boosts the performance.

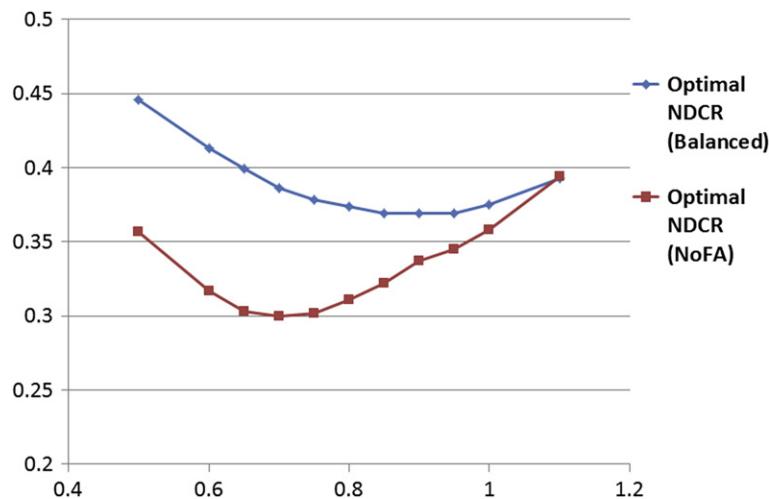
The adopted fusion mechanism combines audio and visual detection scores by weighted summations. [Figure 13.14](#) shows how the video weight  $w_v$  (the audio weight  $w_a = 2.0 - w_v$ ) effects overall performance. It is interesting that for the balanced and the NoFA profiles, different weights should be used to achieve the best overall performance. When  $w_v = 0.9$ , the Optimal NDCR for the balanced profile is 0.369, and when  $w_v = 0.7$ , the Optimal NDCR for the NoFA profile is 0.3. Compared to the best audio- and visual-based performance shown in [Table 13.3](#), the overall performance is significantly improved.

### Performance impact by transformations

[Table 13.4](#) lists detailed performance of the NoFA profile for different types of audio/visual transformations. There are 7 audio transformations and 8 visual transformations, and a brief description of each

**Table 13.3** Optimal NDCR of Audio- and Visual-Based Approaches

Subsystem	Combination	Optimal NDCR (balanced)	Optimal NDCR (NoFA)
Audio	Original	0.601	0.515
	Original and MP3	0.572	0.487
	Original, MP3, and bandpass filtering	<b>0.561</b>	<b>0.483</b>
Video	Pair 1	0.656	0.699
	Pairs 1–2	0.613	0.613
	Pairs 1–4	0.605	0.592
	Pairs 1–6	0.604	0.591
	Pairs 1–8	0.599	0.586
	Pairs 1–10	0.594	0.581
	Pairs 1–12	<b>0.594</b>	<b>0.579</b>

**FIGURE 13.14**

Performance of fused audio and visual detection results.

**Table 13.4** Optimal NDCR for Different Audio and Visual Transformations (NoFA Profile)

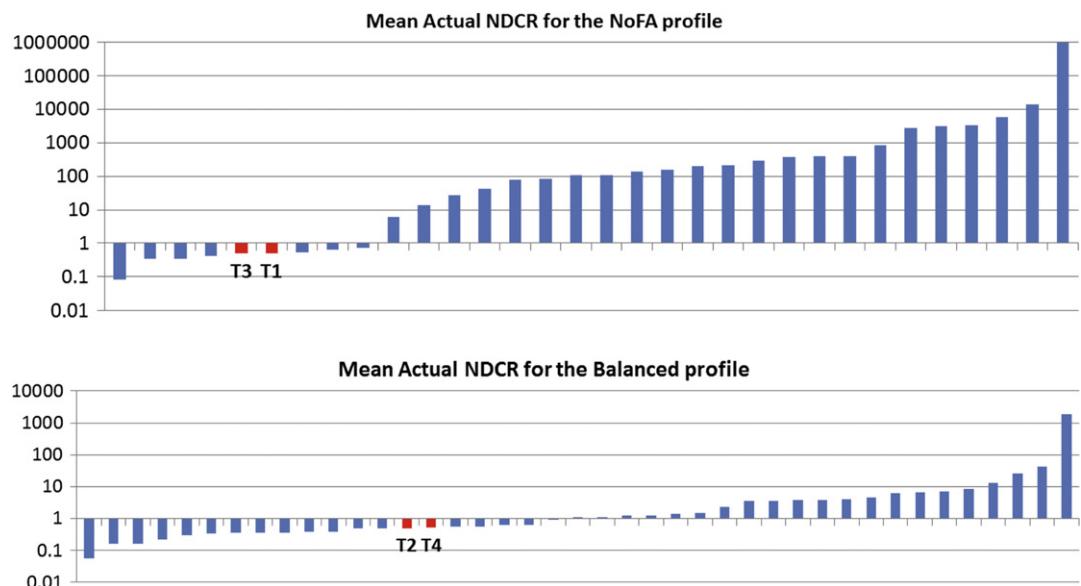
Transformations in query videos		Optimal NDCR
Audio	TA1: nothing	0.277
	TA2: MP3 compression	<b>0.226</b>
	TA3: T2 and multiband companding	0.269
	TA4: bandwidth limit and single-band companding	0.289
	TA5: mixed with speech	0.317
	TA6: T5 and multiband compress	0.316
	TA7: bandpass filter, mixed with speech, compress	0.407
Video	TV1: Simulated camcording	0.377
	TV2: Picture in picture	0.424
	TV3: Insertions of pattern	<b>0.181</b>
	TV4: Strong re-encoding	0.344
	TV5: Change of gamma	0.215
	TV6: Decrease in quality	0.322
	TV8: 3 randomly selected combination of crop, shift, contrast, mirroring, etc.	0.215
	TV10: 3 randomly selected combinations from T2–T5, T6, and T8	0.323

transformation is given in the table. It is clear that the proposed system is robust at detecting audio TA2, and visual TV3, TV5, and TV8 transformations. Audio transformation TA7 and visual transformation TV2 are still challenging for the system.

### Overall performance

On average, the system takes 30 s to process each query, much less than the median processing time from all TV2011 CCD participants (179 s). The average frame level detection accuracy (F1) is 0.89, and it is slightly better than the median accuracy from all participants (0.87). The visual and audio subsystems take 19 and 11 s, respectively, to process each query video, and the time for fusion is negligible. In real applications where not all pairs of transformations have to be considered, the system can perform even more efficiently.

Overall, the system achieves good NDCR performance, significantly better than the median results in all categories. Evaluation results shown in Figure 13.15 indicate that run T2 performs slightly better than run T4 (the NoFA profile), and T3 is better than T1 (the Balanced profile). The system also performs well in the NoFA profile, important in real applications where high false alarm rates are more of a concern. Compared to the top performer in the TRECVID 2011 CCD task, the proposed system is faster: 170 s for the top performer and only 30 s for the proposed system. The primary improvement in speed comes from utilization of only one visual and one audio feature and fast score fusion. These optimizations produced slightly worse performance compared to other systems, with average actual NoFA NDCR of



**FIGURE 13.15**

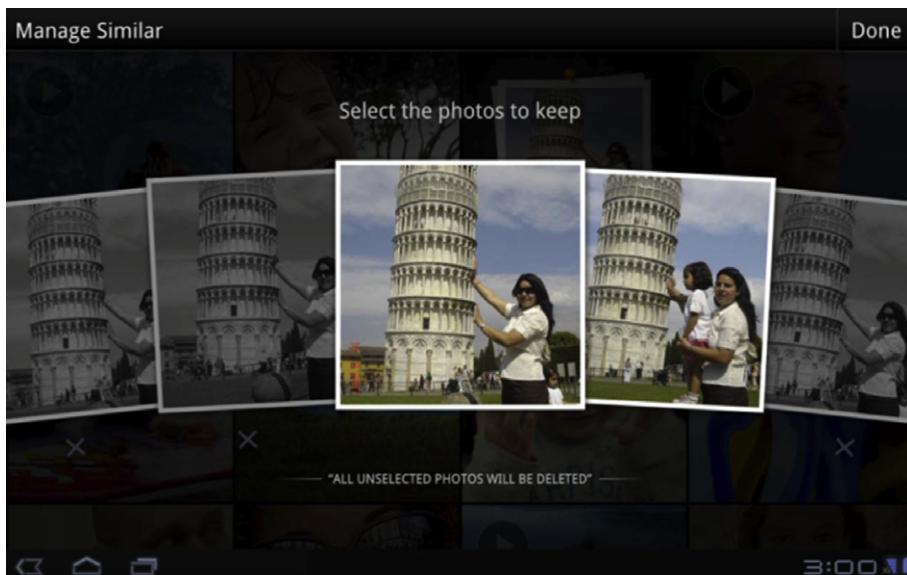
NDCR performance for NoFA and balanced profiles in TRECVID 2011 with proposed system runs marked with labels.

0.33 for the proposed system and 0.08 for the top performer. Choices made in the proposed system best suit certain applications that require low latency yet tolerate less accuracy.

### 5.13.5.3 Personal media organization

Turning now to focus on applications of CCD, today's mobile devices allow users to easily capture high quality media and cloud-based services enable long-term media storage and sharing across devices. As users quickly accumulate a large number of media files, automated tools for organizing that media become essential. While metadata like geo-location and date are available, content-based linking of visually similar images creates innovative capabilities.

Figure 13.16 shows a prototype tablet client for a service called VidCat with the capability to organize personal media with content analysis [65]. Vidcat enables three powerful use cases. First, video segmentation divides content into manageable, syntactically coherent units and facilitates on-line editing and browsing with a minimally redundant set of keyframes. These keyframes can then be analyzed as individual photos in the system. Second, face detection and face similarity metrics as image analysis methods provide validated face regions coupled with face tag suggestions. Third, near-duplicate image detection clusters visually similar photos to go beyond traditional time and heuristic photo grouping to minimize redundant sets of photos. In the case shown in Figure 13.16, the user has uploaded many photos after traveling, and the cloud-based media service has determined that five of the images are of the same subject by matching a majority of image features. The user interface groups these images and



**FIGURE 13.16**

Organizing personal media with visual copy detection.

represents the set by a single icon that the user may touch to expand (as shown in the figure) and choose which of the images to keep.

### 5.13.6 Region and partial content search

Leveraging the proposed visual copy detection algorithms, a robust partial content and region search system was evaluated. The following sections describe enhancements for instance-based search and a working prototype using the proposed copy detection system.

#### 5.13.6.1 Enhancements for TRECVID INS

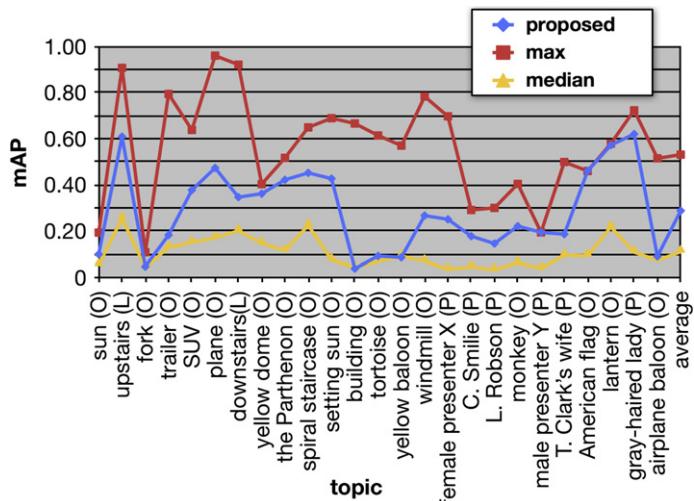
In this chapter, the instance search task (INS) was interpreted as a subset of general copy detection, where queries are regions and small visual objects instead of full scenes after transformations. Other works have focused on textual or mid-level visual semantics but experiments with these methods in prior evaluations provided negligible gains compared to a CCD baseline. The visual module utilizes all local features extracted from the entire keyframe, but different challenges must be addressed for regions. Is sparse or dense local feature sampling best for regions? Should region-level search include the surrounding image context of the query region? In the proposed system, empirical studies found dense sampling produced too many features, increasing resource demands and weakening geometric verification. Similarly, context improved search performance for smaller and homogenous regions.

#### 5.13.6.2 TRECVID INS evaluation criteria

The instance search task was evaluated as a search task in 2010 and 2011 as part of TRECVID. The core metric for evaluation is mean average precision (mAP) at a depth of 1000 results, which favors systems that return many correct results near the top of the list. As a pilot task, the test dataset was smaller than the CCD task, but still produced 109135 video subshots from 84128 reference videos after analysis with the shot boundary component described above. A total of 25 queries were evaluated independently to produce lists of 1000 shots that were evaluated by NIST. In submissions created for TRECVID 2011, no audio or textual features were considered during indexing or query evaluation and no visual transformations or normalizations were applied. Additionally, all results were computed with the multi-query normalization from [Section 5.13.4.2](#) followed by average fusion.

#### 5.13.6.3 TRECVID INS evaluation results

Top performance, median performance, and performance of the proposed system for the INS TRECVID 2011 task are illustrated in [Figure 13.17](#). Each topic is annotated with a letter that denotes whether the query was an object (O), location (L), or person (P). Inspection of this figure indicates that the proposed system generally performed better than the median submission. Also, queries with large objects (*SUV*, *the Parthenon*) generally performed well while smaller objects (*fork*, *airplane balloon*, *yellow balloon*) generally exhibited poor performance. As discussed in [Section 5.13.6.1](#) and empirically proven, sparse local features are often insufficient for full representation of small objects. Even though scene context was included for INS queries, not enough features were common among these query targets for consistent retrieval. Similarly, while inclusion of scene context aided queries for locations and people, it hurt

**FIGURE 13.17**

Mean average precision of proposed INS run and peers.

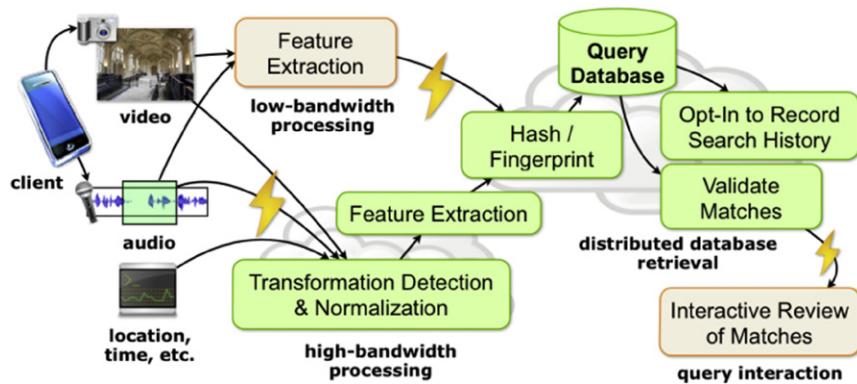
queries for large objects that have unremarkable textures (*building, tortoise*). Almost all local features that are sparsely sampled use an interest operator to determine adequate sample locations in an image so areas of smooth, low-contrast texture are seldom selected for feature points.

These evaluation results demonstrate that the proposed system is quite suitable for object-level copy detection. Future work for object-level copy detection focuses on a more precise determination of the query object through saliency operators, region-based re-ranking of results to complement strict geometry verification, and exploring methods for complementary sampling of dense and sparse local features.

#### 5.13.6.4 Mobile content-based copy detection

Image and audio services discussed earlier harness automatic content recognition to link users to products, content, and social sharing sites. To showcase the proposed copy detection capabilities, an iOS and Android client application called MediApriori was created with a web-based interface to a back-end for product search. Currently, the system has collected over 200 thousand of product images from the web and over 4.5 million keyframes of TV programs captured from the over the air TV broadcast. The MediApriori application in Figure 13.18 allows the capture of audio streams and images that are either processed locally or immediately submitted to the back-end server as a query.

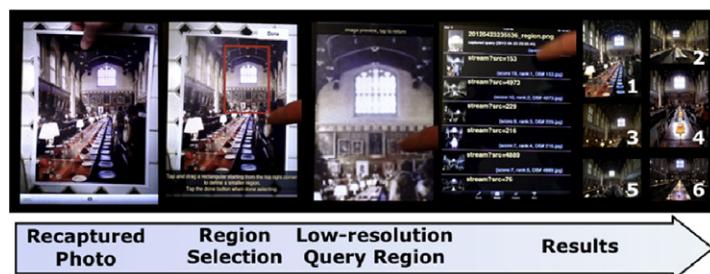
Again, an important problem to consider for systems using copy detection technology on mobile devices is whether or not to include feature extraction on-device. Recent work has demonstrated that poorly chosen feature computation methods can require the transmission of data requiring more resources than the original image [50]. To reduce the complexity of the MediApriori interface and allow feature extraction methods to be easily updated as algorithms improve, most experiments with the proposed system extract features on the back-end (high-bandwidth processing). A second problem for large-scale retrieval with mobile devices is degraded capture quality that includes extreme noise

**FIGURE 13.18**

Overview of mediapriori mobile copy detection app.

conditions in audio and out-of-plane video transformations that are uncharacteristic for normal digital consumption. Recent work focused on copy detection of images characterized this problem as domain adaption and proposed a method for projecting features into a manifold for better recognition [57]. Experimentally, the proposed system was often able to overcome this problem with two solutions. First, the introduction of more instances of the same query object (i.e., two or more views) accommodated viewpoints that previously matched with too few local features. Second, the inclusion of additional normalization steps and a stronger bias for local features that better represent object-centric queries, as in the INS task. Figure 13.19 illustrates one region-based query from a printed and recaptured photo and the diverse lighting and view results that were achieved with the MediApriori system.

Systems deploying copy detection technology on mobile devices also have challenges that the proposed system helps to overcome. One challenge for large-scale mobile retrieval is degraded capture quality including extreme noise conditions in audio and out-of-plane video transformations uncommon in normal digital consumption. The proposed system solves this challenge with normalizations and transformations included in the visual and audio detection modules. Another challenge is the effective transmission of features from on-device extraction, which recent work has shown can be larger than the

**FIGURE 13.19**

Region-based query and results in mediapriori.

original image. Proposed indexing optimizations like hash-based pruning of local features and audio fingerprints can be sent to the device to eliminate low quality hashed features. Finally, applications (like product search) with multiple views and diverse object appearances are supported by the large-scale robustness of the near-duplicate task and multi-query and context optimizations for region-based search.

### 5.13.7 Conclusion and future trends

This chapter presented a review on the recent progress of large-scale video copy detection methods. Most systems utilize both audio and visual information to solve this challenging problem. Focusing on the system developed in AT&T Labs, performance of its underlying algorithms as well as prototype applications built on it were reported. Both audio fingerprint and visual hash values are extracted to compare the video content, and the overall results are determined based on the fusion of both audio and visual query scores. The effectiveness and efficiency of this system has been demonstrated by its strong performance in TRECVID content-based copy detection and instance-based search tasks and its utility in consumer-oriented services for content organization and product matching.

As a research field, video copy detection has been active for the last decade. Many novel audio- and visual-based methods have been proposed and tested on common evaluation dataset. They undoubtedly help mature this field and enable many large-scale multimedia applications and services. While all these achievements are encouraging and promising, there are still many challenges in this area desiring further investigation. Following are some potential research trends from the authors' perspective.

- Scalability: With more multimedia data created daily, the video copy detection engine is facing an unprecedented scale issue. There are billions of images and millions of hours of audio and video content on the web, and these numbers are growing at an increasing speed. Indexing and searching them requires very efficient algorithms in terms of both computational complexity and the storage requirement. Being part of the big data era, further improvements are needed to address scalability of the core video copy detection algorithm.
- Mobility: Mobile computing platform revolutionizes the manner in which people live, work, entertain, and communicate. Mobile applications with video content search capability will meet the users' instant needs for desired information. The challenge here is how to create a user-friendly yet powerful mobile app that delivers a seamless and personalized experience for the users with the integration of the user's context, for example, time, location, environment, etc.
- Fusion from multiple cues: The current fusion scheme mainly involves audio and visual information. Textual, linguistic, facial, and semantic information can be extracted from the embedded audio and visual streams, and they can further improve the video copy detection performance.

---

## References

- [1] B. Girod, V. Chandrasekhar, D. Chen, N. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, R. Vedantham, Mobile visual search, *IEEE Signal Process. Mag.* 28 (4) (2011) 61–76.
- [2] W. Hu, N. Xie, L. Li, X. Zeng, S. Maybank, A survey on visual content-based video indexing and retrieval, *IEEE Trans Syst. Man Cybern. Part C: Appl. Rev.* 41 (6) (2011) 797–819.

- [3] S. Antani, R. Kasturi, R. Jain, A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video, *Pattern Recognit.* 35 (2002) 945–965.
- [4] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, F. Stentiford, Video copy detection: a comparative study, in: ACM International Conference on Image and Video Retrieval (CIVR) 2007, Amsterdam, Netherlands, July 9–11, 2007.
- [5] O. Chum, J. Philbin, M. Isard, A. Zisserman, Scalable near identical image and shot detection, in: ACM International Conference on Image and Video Retrieval (CIVR) 2007, Amsterdam, Netherlands, July 9–11, 2007, pp. 549–556.
- [6] X. Wu, C. Ngo, A. Hauptmann, H. Tan, Real-time near-duplicate elimination for web video search with content and context, *IEEE Trans. Multimedia* 11 (2) (2009) 196–207.
- [7] M. Hill, J. Smith, Design and evaluation of an effective and efficient video copy detection system, in: IEEE International Conference on Multimedia and Expo (ICME), 2010, Singapore, July, 2010, pp. 19–23.
- [8] J. Li, Y. Liang, B. Zhang, Video copy detection based on spatiotemporal fusion model, *Tsinghua Sci. Technol.* 17 (1) (2012) 51–59.
- [9] L. Rabiner, B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
- [10] L. Lu, H. Zhang, H. Jiang, Content analysis for audio classification and segmentation, *IEEE Trans. Speech Audio Process.* 10 (7) (2002) 504–516.
- [11] A. Wang, An industrial-strength audio search algorithm, in: International Conference on Music Information Retrieval (ISMIR), Washington, DC, USA, 2003.
- [12] R. Typke, F. Wiering, C. Veltkamp, A Survey of music information retrieval systems, in: International Conference on Music Information Retrieval (ISMIR), London, UK, September 11–15, 2005, pp. 153–160.
- [13] P. Cano, E. Batlle, T. Kalker, J. haitsma, A review of audio fingerprinting, *J. VLSI Signal Process.* 41 (2005) 271–284.
- [14] R. Roopalakshmi, G. Reddy, A novel approach to video copy detection using audio fingerprints and PCA, *Procedia CS* (2011) 149–156.
- [15] Y. Wang, Z. Liu, J. Huang, Multimedia content analysis using audio and visual information, *IEEE Signal Process. Mag.* (2000) 12–36.
- [16] P. Atrey, M. Hossain, A. Saddik, M. Kankanhalli, Multimodal fusion for multimedia analysis: a survey, *Multimedia Syst.* 16 (2010) 345–379.
- [17] A. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and TRECVID, in: ACM Multimedia Information Retrieval Workshop (MIR) 2006, Santa Barbara, California, USA, October 26–27, 2006.
- [18] M. Jiang, S. Fang, Y. Tian, T. Huang, W. Gao, PKU-IDM@TRECVID2011 CBCD: Content-Based Copy Detection with Cascade of Multimodal Features and Temporal Pyramid Matching, TRECVID 2011 Workshop, Gaithersburg, MD, December 5–7, 2011.
- [19] V. Gupta, P. Varcheie, L. Gagnon, G. Boulian, CRIM at TRECVID 2011: Content-Based Copy Detection Using Nearest-Neighbor Mapping, TRECVID 2011 Workshop, Gaithersburg, MD, December 5–7, 2011.
- [20] M. Ayari, J. Delhumeau, M. Douze, H. Jegou, INRIA@TRECVID'2011: Copy Detection & Multimedia Event Detection, TRECVID 2011 Workshop, Gaithersburg, MD, December 5–7, 2011.
- [21] M. Douze, H. Jegou, C. Schmid, An image-based approach to video copy detection with spatio-temporal post-filtering, *IEEE Trans. Multimedia* 12 (4) (2010) 257–266.
- [22] Z. Liu, E. Zavesky, B. Shahraray, N. Zhou, AT&T Research at TRECVID 2011, TRECVID 2011 Workshop, Gaithersburg, MD, December 5–7, 2011.
- [23] H. Zhang, K. Kankanhalli, S. Smoliar, Automatic partitioning of full-motion video, *ACM Multimedia Syst.* 1 (1) (1993) 10–28.
- [24] B. Yeo, B. Liu, Rapid scene analysis on compressed video, *IEEE Trans. Circ. Syst. Video Technol.* 5 (6) (1995) 533–544.

- [25] B. Shahraray, Scene change detection and content-based sampling of video sequences, in: *Digital Video Compression: Algorithms and Technologies, Proceedings of SPIE*, vol. 2419, 1995.
- [26] P. Over, G. Awad, W. Kraaij, A. Smeaton, TRECVID 2007—Overview, *TRECVID 2007 Workshop*, Gaithersburg, MD, November 5–6, 2007.
- [27] A. Smeaton, P. Over, A. Doherty, Video shot boundary detection: seven years of TRECVID activity, *Comput. Vis. Image Understand.* 14 (4) (2010) 411–418.
- [28] Z. Liu, D. Gibbon, E. Zavesky, B. Shahraray, P. Haffner, A fast, comprehensive shot boundary determination system, in: *IEEE International Conference on Multimedia and Expo (ICME)*, 2007, pp. 1487–1490.
- [29] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, *Int. J. Comput. Vis.* 42 (3) (2001) 145–175.
- [30] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, C. Schmid, Evaluation of GIST descriptors for web-scale image search, in: *ACM International Conference on Image and Video Retrieval (CIVR)* 2009, Santorini, Greece, July 8–10, 2009.
- [31] M. Stricker, M. Orengo, Similarity of color images, in: *Proceedings of SPIE, Storage and Retrieval for Image and Video Databases III*, vol. 2420, 1995, pp. 381–392.
- [32] E. Zavesky, Z. Liu, D. Gibbon, B. Shahraray, Searching visual semantic spaces with concept filters, in: *IEEE International Conference on Semantic Computing (ICSC)*, Irvine, California, USA, September 17–19, 2007.
- [33] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8) (1996) 837–842.
- [34] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, *Foundations Trends Comput. Graph. Vis.* 3 (3) (2007) 177–280.
- [35] E. Tola, V. Lepetit, P. Fua, DAISY: an efficient dense descriptor applied to wide-baseline stereo, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (5) (2010) 815–830.
- [36] C. Harris, M. Stephens, A combined corner and edge detector, in: *Proceedings of Fourth Alvey Vision Conference*, 1988, pp. 147–151.
- [37] D. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis. (IJCV)* 2 (60) (2004) 91–110.
- [38] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: *European Conference on Computer Vision*, 2006, pp. 430–443.
- [39] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable external regions, *Br. Mach. Vis. Comput.* 22 (10) (2004) 761–767.
- [40] H. Bay, T. Tuytelaars, L. Gool, SURF: speeded up robust features, *Comput. Vis. Image Understand. (CVIU)* 110 (3) (2008) 346–359.
- [41] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (10) (2005) 1615–1630.
- [42] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua, BRIEF: computing a local binary descriptor very fast, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (7) (2012) 1281–1298.
- [43] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: *The IEEE International Conference on Computer Vision (ICCV)* 2011, Barcelona, Spain, November 6–13, 2011, pp. 2564–2571.
- [44] S. Leutenegger, M. Chli, R. Siegwart, BRISK: binary robust invariant scalable keypoints, in: *The IEEE International Conference on Computer Vision (ICCV)*, 2011.
- [45] A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: fast retina keypoint, in: *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA, June 16–21, 2012.
- [46] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, in: *Proceedings of the Conference Computer Vision and Pattern Recognition (CVPR)*, 2004, pp. 511–517.
- [47] M. Heikkila, M. Pietikainen, C. Schmid, Description of interest regions with local binary patterns, *Pattern Recognit.* 42 (3) (2009) 425–436.

- [48] OpenCV, Open Source Computer Vision Library, <<http://www.intel.com/technology/computing/opencv/>>, 2013.
- [49] M. Muja, D. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: International Conference on Computer Vision Theory and Applications (VISAPP) Lisboa, Portugal, February 5–8, 2009, pp. 331–340.
- [50] A. Andoni, P. Indyk, Near-optimal hashing algorithms for near neighbor problem in high dimension, Commun. ACM 51 (1) (2008).
- [51] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: IEEE International Conference on Computer Vision (ICCV) 2003, Nice, France, October 14–17, 2003, pp. 1470–1477.
- [52] J. Yang, Y. Jiang, A. Hauptmann, C. Ngo, Evaluating bag-of-visual-words representations in scene classification, in: The International Workshop on Multimedia Information Retrieval (MIR), September 28–29, 2007, Augsburg, Germany, pp. 197–206.
- [53] C. Tsai, Bag-of-Words representation in image annotation: a review, ISRN Artif. Intell. 2012 (2012), Article ID 376804.
- [54] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2006, New York, NY, USA, June 17–22, 2006, pp. 2161–2168.
- [55] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
- [56] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision, Marseille, France, October 12–18, 2008, pp. 304–317.
- [57] V. Gupta, G. Boulian, P. Cardinal, CRIM's content-based audio copy detection system for TRECVID 2009, in: International Workshop on Content-based Multimedia Indexing, Grenoble, France, June 23–25, 2010.
- [58] M. Mihcak, R. Venkatesan, A perceptual audio hashing algorithm: a tool for robust audio identification and information hiding, in: The Fourth International Workshop on Information Hiding 2001, Pittsburgh, PA, USA, April 25–27, 2001, pp. 51–65.
- [59] J. Haitsma, T. Kalker, A highly robust audio fingerprinting system, in: Third International Conference on Music Information Retrieval 2002, Paris, France, October 13–17, 2002, pp. 107–115.
- [60] S. Baluja, M. Covell, Waveprint: efficient wavelet-based audio fingerprinting, Proc. Pattern Recognit. 41 (2008) 3467–3480.
- [61] X. Anguera, A. Garzon, T. Adamek, MASK: robust local features for audio fingerprinting, in: IEEE International Conference on Multimedia and Expo 2012, Melbourne, Australia, July 9–13, 2012.
- [62] C. Snoek, M. Worring, A. Smeulders, Early versus late fusion in semantic video analysis, in: ACM Conference on Multimedia 2005, Singapore, November 6–11, 2005.
- [63] J. Chen, T. Huang, A robust feature extraction algorithm for audio fingerprinting, in: Pacific-Rim Conference on Multimedia (PCM), Tainan, Taiwan, December 9–13, 2008, pp. 887–890.
- [64] H. Jegou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, Int. J. Comput. Vis. 87 (3) (2010) 316–336.
- [65] L. Begeja, E. Zavesky, Z. Liu, D. Gibbon, R. Gopalan, B. Shahraray, VidCat: an image and video analysis service for personal media management, in: Proc. SPIE 8667, Multimedia Content and Mobile Devices, February 3, Burlingame, CA, USA, 2013.

# Index

## A

Absolute Category Rating (ACR), 232  
Accommodation, 120, 122, 147  
Accurate motion, 28  
Achievable quintuple, 257, 286  
Acoustic features, 437  
Adaptive DPCM (ADPCM) coder, 266  
Adaptive Loop Filter (ALF), 188  
Adaptive Motion Vector Prediction (AMVP), 100  
Adaptive multi-rate wideband (AMR-WB) standard, 267  
Additive Operator Splitting (AOS) scheme, 387  
Ad-hoc networks  
    multi-hop broadcast/flooding in  
        area based flooding, 350  
        message delivering, 349  
    neighbor-knowledge based flooding, 350  
    probabilistic flooding, 350  
    Simple Flooding, 349  
    routing protocols for  
        Ad-Hoc On-Demand Distance Vector, 347  
        Augmented Tree-based Routing Algorithm, 349  
        comparative overview of, 346–347  
    Destination-Sequenced Distance Vector Algorithm, 347  
    Dynamic Source Routing Algorithm, 347  
    Optimized Link-State Routing, 349  
    Temporally-Ordered Routing Algorithm, 347  
    video streaming over  
        CoDiO framework, 352  
        Multiple Description Coding, 351  
        multi-tree construction protocols, 351  
        network-oriented metrics, 351  
        rate-less codes, 351  
        scalability, 351  
        unicast and multicast transmission, 351  
Ad-Hoc On-Demand Distance Vector (AODV), 347  
Advanced Simple Profile (ASP), 21  
Affine motion model, 34  
All Intra (AI), 106  
Analytic wavelets, 46  
Anchor frames, 332  
Angular Error (AE), 83  
Aperture problem, 30–31, 33  
Apparent motion, 28  
Application Level Multicast (ALM), 341  
Approximate nearest neighbor (ANN) search  
    methods, 393  
Arithmetic coding, 14, 132  
Artifact, 222

Artifact-Based Video Metric (AVM), 243  
Audio fingerprint, 438  
Augmented Tree-based Routing Algorithm (ATR), 349  
Auto Regressive (AR) model, 180, 201–203, 265  
Auto Regressive Moving Average (ARMA), 180, 201, 203  
Auto-repeat request (ARQ) process, 296  
Autostereoscopic displays  
    multi-view systems  
        four views display, 125–126  
        seven-view display, 127  
    super multi-view systems, 127–128  
    two-view systems, 125

## B

Backward motion compensation, 134  
Bag-of-Visual-Words (BoW) model, 383, 385, 431–432  
    based multimedia content-based visual retrieval, 385  
Best neighborhood matching method (BNM), 298  
Bi-directional block-based motion compensation  
    general approach used in, 134–135  
    typical group of coded frames, 134–135  
Bi-directional (B) frames, 134, 332–333  
Bidirectional prediction, 15  
Bilinear interpolation (BI), 297, 301  
Bilinear motion model, 35  
Binary Robust Independent Elementary Features (BRIEF), 428  
Binary Robust Invariant Scalable Keypoints (BRISK), 428  
Binary SIFT  
    descriptors  
        feature matches, examples of, 390–391  
        feature vector, 389  
        local matching results, examples, 392  
        scalar quantization strategy, 389  
        statistics on, 390, 392  
    indexing with  
        code word definition and exploration, 403–404  
        code word expansion, 404–405  
     $L_2$ -distance between SIFT features, 388  
    potential concern of, 393  
Bitstream-switching process, 339  
Bjontegaard Delta (BD), 189  
Bjontegaard Delta Mean Opinion Score (BD-MOS) metric, 106  
Bjontegaard Delta PSNR (BD-PSNR) metric, 106  
Block-based motion-compensated video coding

- approach used in, 134  
 basic block diagram of, 133  
 picture types, 17  
 prediction modes, 15  
 video decoder  
     operation of, 19  
     structure, 19  
 video encoder, operation of  
     generic structure of, 17–18  
     inter-mode encoding, 18  
     intra-mode encoding, 17–18
- B**  
 Block-based video coding techniques  
     framework, 199–201  
 non-parametric approaches, 206  
     general template matching algorithm, 207  
     optimization strategies and performance, 207  
 parametric approaches, 201  
     AR-based prediction, 202  
     ARMA-based prediction, 203  
     H.264/AVC and HEVC, 201  
     implementation of, 201  
     LS-based linear prediction, 201–202  
     optimization strategies and performance, 204  
 PDE-based approaches, 205  
     intra prediction based on Laplace PDE, 205  
     intra prediction based on TV, 206  
     optimization strategies and performance, 206–207  
 perception-oriented coding techniques, 201  
 Block level quantization step, 176  
 Block matching approaches  
     candidate motion vector, 47  
     3D recursive search technique, 65  
     flexible motion partitioning technique, 65  
     forward motion estimation, 47, 49  
     matching criterion, 55  
     correlation-based criteria, 60  
     illumination-invariant criterion, 59  
     norm-based criteria, 56, 58–59  
         for wavelet-based compression, 61  
     matching-primitive techniques, 47  
     search window and search strategy, 52  
     conjugate directions search, 51  
     diamond search, 54–56  
     2D-logarithmic search, 51, 53  
     fast search, 50  
     full search, 49  
     hexagon search, 55–56, 58  
     three-step search, 52, 54  
     sub-pixel accuracy, 61–64  
     template matching technique, 66, 70  
     variable-size block matching techniques, 63, 66  
     versions of, 49
- Boundary matching algorithm (BMA), 304  
 Boundary matching error (BME), 306–307  
 B (Bi-predictive) picture, 95  
 Brewster’s stereoscope, 129  
 Broadband channels, 325  
 B (Bi-predictive) slice, 96  
 Bwana Devil (3D Cinema), 130
- C**  
 Cafforio-Rocca algorithm, 40  
 Candidate motion vector, 47  
 Canny edge detection algorithm, 422  
 Cepstral coefficients (CC), 437  
 Clustering techniques, 393  
 Code word  
     definition and exploration, 403  
     expansion, 405, 404  
 Coding tree unit (CTU), 9, 100, 176  
     picture partitions, 101  
     size, 110  
     split, 101–103  
 Coding unit (CU), 101–102  
 Cognitive Radio (CR), 284  
 Common Intermediate Format (CIF), 95, 97  
 Common test conditions, 106  
 Communication and cooperation between encoder  
     and decoder, 352, 355  
 Complexity *vs.* cost, 5  
 Computer generated imagery (CGI), 144  
 Computer Research Institute of Montreal  
     (CRIM), 418  
 Concealment mode selection algorithm, 312  
 Congestion-Distortion Optimization (CoDiO), 352  
 Content-based copy detection (CCD), 418  
     mobile, 450  
     TRECVID  
         evaluation results, 444  
         task and evaluation criteria, 443  
 Content-based retrieval. *See* Multimedia content-based visual retrieval  
 Content-based visual search, 383  
 Content Delivery Network (CDN), 338  
 Content Distribution Networks (CDNs), 340  
 Context Adaptive Binary Arithmetic Coding  
     (CABAC), 97, 100, 114  
 Contrast Sensitivity Function (CSF), 162, 167, 193  
 Cooperative networking (CoopNet), 343–344  
 Copydays dataset, 383  
 Copy location accuracy, 444  
 Correlation analysis, 238  
 Cross-search algorithm, 52

**D**

- DAISY (local visual features), 427–428
- Data format, 131, 148
- 3D display technologies
  - direct-view technologies, 123
  - autostereoscopic displays (*see* Autostereoscopic displays)
  - stereoscopic displays, 123
  - goal of, 119
  - head-mounted technologies, 123
  - requirement of, 123
  - volumetric displays, 123
- Decoding buffer, 335
- Degradation Category Rating method (DCR), 233
- De-jittering buffer, 335
- Delay *vs.* performance, 5
- Delta-sigma quantization (DSQ), 280
- Depth Enhanced Stereo (DES) format, 142
- Depth estimation
  - active depth sensing, 143
  - 2D-3D conversion from single view, 144
  - from stereo matching, 143
  - synthetic views and depth maps from CGI, 144
- Depth image-based rendering (DIBR), 137
- Depth map coding
  - edge-adaptive wavelets, 145
  - graph-based transforms, 146
- Destination-Sequenced Distance Vector Algorithm (DSDV), 347
- Diamond search (DS) algorithm, 54–56
- Difference Mean Opinion Scores (DMOS), 234, 240
- Difference of Gaussian (DoG), 386, 428
- Differential Pulse Code Modulation (DPCM), 20, 184, 265
- Diplopia, 121
- Directional entropy, 298, 301
- Directional interpolation (DI), 298, 300
- Discrete Cosine Transform (DCT), 11–12, 20, 28, 95, 159, 163, 192, 253
- Discrete Sine Transform (DST), 100
- Discrete Wavelet Transform (DWT), 28
- Displaced Frame Difference (DFD), 15, 18, 45
- Display
  - 3D technologies, 123
  - volumetric, 123
  - autostereoscopic (*see* Autostereoscopic displays)
  - stereoscopic, 122–123
  - distortion caused by, 148
- 2D-logarithmic search, 51–53
- Double stimulus methods, 233
- Double Stimulus Continuous Quality Scale (DSCQS), 233
- Double Stimulus Impairment Scale (DSIS) method, 106, 233

- 3D recursive search (3D RS) block matching technique, 65
- DupImage dataset, 383
- Dynamic Source Routing Algorithm (DSR), 347

**E**

- Embedded Zero-Tree Wavelet (EZW) coder, 251
- Entropy coding, 132, 192
  - data compression technique, 132
  - in H.264/AVC, 97
  - inter-frame prediction, 133–134
- Entropy-constrained multiple description scalar quantizers (ECMDSQs), 255, 263
- Entropy decoding, 19
- Entropy encoding, 14
- Error concealment (EC), 353
  - defined, 296
  - non-normative post-processing, 295
  - primary types
    - spatial error concealment (*see* Spatial error concealment (SEC) method)
    - temporal error concealment (*see* Temporal error concealment (TEC) method)
- Error in flow Endpoint (EE), 84
- Error resilience, 284, 296
  - vs.* redundancy, 5
  - trade-off between coding efficiency and, 279
- Expanding Window Network Coding (EWNC), 368
- External boundary matching error (EBME), 304, 306–307

**F**

- Fast Retina Keypoint (FREAK), 428–429
- Fast search techniques, 50
- Feature matching, 387, 390–391
  - between images, problem of, 403
  - recall rate of, 394, 396
- Feature quantization
  - scalable cascaded hashing, 396
  - cascaded hashing, 397, 400
  - dimension reduction on SIFT by PCA, 397–399
  - matching verification, 401
  - vector quantization, 393
  - issues in, 394
  - visual word expansion scheme, 395
- Feature representation. *See* Local feature representation
- Features from Accelerated Segment Test (FAST), 427
- Finite state machine (FSM), 421
- Flexible macroblock ordering (FMO), 296, 317
- Forward Error Correction (FEC), 336–337, 353
- Forward motion compensation, 134

Forward motion estimation, 47, 49  
 Fountain codes, 351  
 Fourier/DCT domain, 43  
 Frame Compatible Stereo, 131  
 Frame rate conversion, 28  
 Free-viewpoint TV (FTV), 131  
 Friese-Green, William, 130  
 Full reference methods, 240

**G**

Generalized MDC (GMDC), 269  
 Geometric error, 145  
 Geometric min-hashing technique, 386  
 Geometric verification  
     affine estimation, enhancement based on, 410–411  
     with coding maps, 411  
     geometric fan coding (GFC), 407  
     geometric ring coding (GRC), 408  
 Global Motion Compensation (GMC), 78, 80, 212, 219  
 Global Motion Estimation (GME), 79  
 Global spatial quality assessment (GSQA), 216  
 Global temporal quality assessment (GTQA), 217  
 Global visual features  
     categories of, 424  
     color moments, 425  
     edge direction histogram, 426  
     Gabor texture, 425  
     improved, 427  
 Google Similar Image Search, 384  
 Gradient Location and Orientation Histogram (GLOH), 428  
 Graph-based transform (GBT), 146  
 Group of Pictures (GOP), 17, 106, 210, 213  
     four-description, 281–282  
     structure, 17, 332–333

**H**

H.261, 20, 95, 198  
 H.263, 95  
 Haar filters, 44  
 Harris corner detector, 427–428  
 Hashing techniques, 386, 394  
 H.264/AVC, 74  
     aim of, 96  
     distribution of HD video, 94  
     entropy coding, 97  
     interim processing step, 97  
     international standard, 94  
     intra modes in, 331  
     macroblock-based coding structure, 96  
     macroblock partitioning, 96–97

motion compensation in  
     macroblocks and partitions, 74  
     motion vector coding, 77  
     multiple references and generalized P/B frames, 75  
     rate-constraint lagrangian motion estimation, 76  
 variable block size, 76, 78  
 picture slices, 96  
 quarter-pixel accuracy, 63  
 variable-size block matching technique, 65  
 video compression standard, 94, 101  
     vs. HEVC, 107–109  
 HD television (HDTV), 94  
 Helmholtz tradeoff estimator, 69  
 Hexagonal search algorithm (HS), 55–56, 58  
 Hierarchical Network Coding (HNC), 368  
 High-capacity broadband connections, 328  
 High-Definition (HD), 94  
 High Definition Television (HDTV), 155  
 High Efficiency 10-bit (HE10), 188  
 High Efficiency Video Coding (HEVC), 198  
     basic principles, 100  
     picture partitions (*see* Picture partitions)  
     transforms and  $4 \times 4$  transform skipping, 103–105  
     video compression standards comparison, 100–101  
 delta QP (dQP), 189  
 development of, 94  
 HD video broadcasting, 94  
 international standards, 94  
 interpolation filters, 63  
 ITU-T and ISO/IEC, developed by, 114  
 performance evaluation, 106  
     objective quality evaluation (*see* Objective quality evaluation)  
     subjective quality evaluation, 106  
 screen content, 94  
 standard, 156  
 ultra high-definition resolution, 114  
 Ultra High-Definition Television, 94  
 variable-size block matching technique, 65  
 video compression (*see* Video compression)  
 H.262/MPEG-2 Video (MPEG-2), 95  
 Holidays dataset, 383  
 Horn-Schunck algorithm, 37  
 Horopter, 121  
 HTTP streaming protocols, 338–339  
 Huffman coding, 14, 132  
 Human visual system (HVS), 29, 192, 228  
     based quality metrics, 242  
     properties of, 156  
     sensitivity modeling, 162  
         contrast masking, 165  
         frequency masking, 162  
 JND, evaluation of, 170, 174

luminance masking, 164, 171  
 in spatio-temporal, combination of masking, 170  
 temporal masking, 167

**I**

*IEEE Information Theory Workshop* (1979), 255  
 Image analysis and completion video coding  
 block-based scheme, 181  
 IAC approach, 180  
 non-parametric image completion, 181  
 parametric image completion, 180  
 PDE-based image completion, 180–181  
 region-based IAC-VC schemes, 181–182  
 Image Analysis and Completion-Video Coding (IAC-VC), 180  
 Infrastructure network, 345  
 Initial Training Network (ITN), 200  
 Instance-based search (INS), 418  
 Instance search task (INS), 418, 449  
 TRECVID  
 enhancements for, 449  
 evaluation criteria, 449  
 evaluation results, 449–450  
 Instantaneous decoder refresh frames (IDR), 296  
 Instituto Superior Técnico-Perceptual Video Codec (IST-PVC), 179  
 Intensity Dependent Quantization (IDQ), 183  
 Intensity Dependent Spatial Quantization (IDSQ), 185  
 International Standards Organization (ISO), 20  
 International Telecommunications Union (ITU), 94  
 International Telecommunications Union (ITU-T), 20–21, 95  
 Internet  
 evolution of, 328  
 3G/4G and WiFi networks, 253  
 multimedia traffic, 253  
 volume of traffic, 329  
 Interpolation Error (IE), 84  
 Inter-stream synchronization, 334  
 Inter-view prediction approach, 136–137  
 Intra-band masking effect, 165  
 Intra (I) frames, 134, 332  
 Intra-stream synchronization, 334  
 Inverse spatial transform (IDCT), 331  
 I (Intra) picture, 95  
 I (Intra) slice, 96  
 ISO/IEC  
 Moving Picture Expert Group (MPEG), 94–95, 156  
 23002-3 specification, 140  
 Iterative coding of multiple descriptions (ICMD), 284  
 Iterative quantization method, 394  
 ITU-T Video Coding Expert Group (VCEG), 94, 156

**J**

JND-Foveated model (JNDF), 179  
 Joint audio and visual processing. *See also* Video copy detection  
 early fusion scheme, 440–441  
 late fusion scheme, 440–441  
 multi-query result normalization and fusion, 442  
 Joint Collaborative Team on Video Coding (JCT-VC), 94, 188  
 Joint source and channel coding (JSCC), 353  
 Just Noticeable Distortion (JND), 159, 161  
 assessing, 170  
 example of, 167  
 integration of  
 perceptual video codecs (*see* Perceptual video codecs (PVCs))  
 in video coding architectures (*see* Video coding)  
 modeling of HVS sensitivity measures (*see* Human visual system (HVS))

**K**

Karhunen-Loëve transform, 255  
 Kernelized locality sensitive hashing method, 394  
 Killer applications, 328  
 Kurtosis coefficient, 234

**L**

Laboratory for Image and Video Engineering (LIVE) database, 237  
 Lapped orthogonal transform (LOT), 256, 270  
 Large diamond-shape-partner (SDSP), 54–55  
 Layered video coding. *See* Scalable video coding  
 Least Mean Square (LMS) technique, 67  
 Least Median of Squares (LMeds), 69  
 Least Squares (LS), 201  
 Least-Trimmed Squares (LTS), 69  
 L-estimators, 69  
 Letterbox detection, 422  
 Linearly independent descriptions (LIDs), 252  
 Linear Predictive Coding (LPC) coefficients, 438  
 Local Binary Patterns (LBP), 428  
 Local feature representation. *See also* Binary SIFT  
 binary local feature, 387  
 binary signature, SIFT, 388  
 generation, 389–390, 392  
 $L_2$ -distance distribution of SIFT matches, 388  
 floating-point local feature, 386  
 Locality sensitive hashing (LSH), 430–431, 438  
 Local Motion Compensation (LMC), 78  
 Local Motion Estimation (LME), 79

- Local Spatial Quality Assessment (LSQA), 216, 218  
 Local Temporal Quality Assessment (LTQA), 216  
 Local visual features  
     DAISY feature, 427  
     feature descriptor, 428  
     keypoint detector, 427  
     stages of, 427  
 Logie Baird, John, 130  
 Loose buffer models, 335  
 Loss-aware rate-distortion optimization (LA-RDO), 355  
 Lossy vs. lossless compression, 5  
 Low Delay with B slices (LDB), 106  
 Lucas-Kanade method, 37
- M**
- Macroblock (MB), 9–10, 175  
     4:2:0 format, 9  
     missing (*see* Video error concealment methods)  
 Manifest file, 339  
 Markov Random Field (MRF) theory, 202  
 Masked Audio Spectral Keypoint (MASK) fingerprint  
     extraction method, 438  
 Matching pursuits (MP) video coding, 273  
 Maximally Stable External Region (MSER), 427  
 MD-balanced rate-distortion splitting (MD-BRDS), 274  
 MD channel-optimized trellis-coded quantization  
     (MDCOTCQ), 264  
 MD predictive-vector quantizers (MD-PVQs), 268  
 MD Spherical quantization with repetition coding of  
     the amplitudes (MDSQRA), 268  
 MD spherical trellis-coded quantization (MDSTCQ), 268  
 MD trellis-coded quantizers (MDTCQ), 264  
 Mean Absolute Error (MAE), 43, 57  
 Mean average precision (mAP), 384  
 Mean Opinion Score (MOS), 171, 233  
 Mean Square Error (MSE), 43, 57, 157, 199  
 MediApriori system, 451, 450  
 Medium Access Control (MAC), 346  
 Mel-frequency cepstral coefficients (MFCC), 437  
 M-estimators, 69  
 Min-hashing technique, 386, 394  
 Mobile ad-hoc network (MANET), 267, 345  
 Mobile content-based copy detection, 450–451  
 Modern hybrid encoder, 331  
 Mode selection method  
     ingredients, 309  
     MCTA and SA, measurement of, 313–314  
     method, 310, 313  
     performance of, 314  
     variations, 312  
 Modified MDSQ (MMDSQ) scheme, 263  
 Modulated Complex Lapped Transform (MCLT), 438  
 Most Apparent Distortion model (MAD), 242  
 Most significant bits (MSBs), 271  
 MOtion-based Video Integrity Evaluation (MOVIE), 179  
 Motion-compensated prediction (MCP), 15  
 Motion compensated temporal activity (MCTA), 312  
     measurement of, 313–314  
 Motion-compensated temporal interpolation, 360  
 Motion compensation (MC), 15, 192  
     motion estimation, 73  
         global, 78  
         in H.264/AVC (*see* H.264/AVC)  
         overlapped block, 78  
         sprites, 81–82  
 Motion constraint, 32  
 Motion estimation (ME), 331  
     block-based motion-compensated video coding architecture  
         (*see* Block-based motion-compensated video coding  
         architecture)  
     block matching approaches  
         candidate motion vector, 47  
         3D recursive search technique, 65  
         flexible motion partitioning technique, 65  
         forward motion estimation, 47, 49  
         matching criterion, 55  
         matching-primitive techniques, 47  
         search window and search strategy (*see* Search window)  
         sub-pixel accuracy, 61–64  
         template matching technique, 66  
         variable-size of, 63  
         versions of, 49  
     FD and DFD frames, pdf of pixel values of, 15–16  
     inter-frame redundancy, 15  
     motion compensated prediction, 15  
     motion compensation, 73  
         GMC, 78, 80  
         in H.264/AVC (*see* H.264/AVC)  
         OBMC, 78–79  
         sprites, 81–82  
     motion representation and models  
         aperture problem, 30–31  
         brightness constancy motion model, 31  
         2D motion vector field and optical flow, 29–30  
         optical flow equation, 31, 34  
         parametric motion models, 33  
         region of support, 35  
         video coding viewpoint, 36  
     motion vector candidates formation, 304  
     multi-resolution approaches  
         advantage of, 71  
         block matching technique, 72–74  
         coarse-to-fine processing, 71  
         dual representation, 71–72

- low-pass pyramid concept, 69, 71
- origin, 69
- objective of, 28
- optical flow approaches
  - dense motion vector field, 38
  - disadvantages, 38
  - Horn-Schunck algorithm, 37
  - Lucas-Kanade method, 37
  - on-line databases, 39
- parametric
  - direct, 67
  - indirect, 67
- performance assessment, 83
  - of optical flow techniques, 83
  - for video coding, 84
- pixel/pel-recursive approaches
  - advantages of, 42
  - Cafforio-Rocca algorithm, 40
  - DFD, definition of, 39
  - Netravali-Robbins method, 39
  - remarks on, 42
- purpose of, 28
- robust estimation, 69
- role of, 85
- transform-domain approaches, 42
  - in Fourier/DCT domain, 43
  - in wavelet domain, 43
- Motion parallax, 122
- Motion vectors (MVs), 296, 353
- Moving Picture Experts Group (MPEG), 21
- MPEG-4, 21, 96
  - GMC encoder, 79–80
  - sprite coding in, 81–82
  - Verification Model, 79
  - Visual, 94
- MPEG-2 Transport Stream (MPEG-2 TS), 333
- Multi-grid block matching techniques. *See* Variable-size block matching techniques
- Multimedia content-based visual retrieval, 325
  - feature quantization, 393
    - scalable cascaded hashing method, 396
    - vector quantization, 393
  - general pipeline, 385
  - index strategy
    - with binary SIFT, 403
    - with visual word, 402
  - indicators
    - accuracy, 384
    - efficiency, 385
    - memory cost, 385
  - issues in
    - image organization, 384
    - image representation, 384
  - image similarity formulation, 384
  - local feature representation, 386
    - binary local feature, 387
    - binary signature, SIFT, 388
    - floating-point local feature, 386
  - post-processing
    - geometric verification by geometric coding, 407
    - query expansion, 406
    - retrieval scoring, 405
  - Multimedia event detection, 418
  - Multimedia signal processing
    - audio-visual signals processing (*see* Video copy detection)
    - content-based visual retrieval (*see* Multimedia content-based visual retrieval)
    - multimedia streaming (*see* Multimedia streaming)
  - Multimedia streaming, 325
    - error
      - concealing, 353
      - control, 355
      - detection, 352
    - multiple description coding
      - channel splitting, 359
      - quantization, 363
      - transform coding, 360
    - network coding
      - butterfly network problem, 364
      - concept of, 364
      - data dissemination, 367–368
      - Linear network coding, 365
      - Practical Network Coding, 366
    - scalable video coding
      - quality scalability, 357
      - spatial scalability, 356–357
      - temporal scalability, 356
    - transmission errors, 352
    - transport protocols
      - bandwidth adaptation, 337
      - MPEG-2 Transport Stream, 333
      - Real-Time Protocol (RTP), 333
      - synchronization, 334
      - transmission delay, 335
      - transmission errors, 336
    - video coding
      - block level, 331
      - frame level, 332
      - redundancy, 330
      - schemes, 331
    - video streaming
      - architecture, 329–330
      - definition, 329
      - Internet, 328–329
      - media server packetizes, 329
      - on-demand streaming, 329

- video-telephony and videoconferencing services, 329
- on wired networks
  - Application Level Multicast, 341
  - Content Distribution Networks, 340
  - cooperative networking, 343–344
  - Peer-to-Peer (P2P) distribution, 341–342
- on wireless networks
  - ad-hoc networks (*see* Ad-hoc networks)
  - application-driven networking, 345
  - infrastructure network, 345
  - mobile ad-hoc network (MANET), 345–346
- Multi-mode error concealment approach, 315–316
  - performance of, 315–317
- Multiple Description Coding (MDC), 353, 357
  - channel splitting, 359–360
  - quantization, 363
  - transform coding, 360–361
    - correlating, example of, 361
    - definition, 360
    - redundant, example of, 362
  - two-channels, scheme of, 358
- Multiple description coding (MDC)
  - advantage, 253
  - architecture, 254
  - development of, 255
  - image coding
    - balanced MDC scheme, 272
    - creating domain-based descriptions of images, 271
    - decomposition and reconstruction technique, 270
    - generalized MDC, 269
    - MDSQ technique, 270
    - phase scrambling technique, 271
    - pre- and post-processing (JPEG), 270
    - transform coding, 269
  - implementations and patents, 285
  - LOT-DCT basis, 256
  - MDVQ-based image watermarking scheme, 284
  - network coding
    - EWNC and RDO scheduling algorithm, 281
    - four-description scheduling framework, 281–282
    - MANET, 252
    - MDC/NC scheme, 280
    - MRC with inter-layer, combination of, 283
    - practical network coding, 280
    - SDC/NC scheme, 280
    - spatial subsampling schemes, 281
  - originated at Bell Laboratories, 255–256
  - PET method, 284
  - plethora of, 256
  - problem of, 255
  - QIMM watermarking technique, 284
  - single signal source, 253
- speech coding
  - context adaptive MDC system, 266
  - ITU-T Recommendation G.711 PCM speech coder, 267
  - LSP-based MDC method, 266
  - MD spherical trellis-coded quantization, 268
  - MDTC, 265–266
  - multiservice business router, 268
  - packet loss concealment, 268
    - SDVQ-PCM speech coder, 267–268
  - SPIHT based image, 284
  - stereoscopic 3D, 283
  - theoretical basis
    - pairwise correlating transforms, 261
    - rate-distortion analysis, 256
    - redundancy rate-distortion analysis, 256, 260
    - scalar and vector quantizers, 262–263
- video coding
  - adaptive concealment scheme, 276–277
  - adaptive redundancy control, 279
  - based on rate-distortion splitting, 274
  - delta-sigma quantization and, 280
  - four-description framework, 275
  - hierarchical B pictures, 275
  - H.264 standard establishment, 274
  - MDC method, 273
  - MDSVC scheme based on MCTF, 275
  - motion vector analysis, 278
  - mutually refining DPCM (MR-DPCM), 273
  - packet loss performance comparison, 276–279
  - partitioning stage, 274
  - spatial concealment, 277
  - temporal concealment, 276–277
- Multiple description PAC (MDPAC), 266
- Multiple Description Quantization (MDQ), 363
- Multiple description scalar quantizer (MDSQ), 255, 262–263
- Multiple description transform coder (MDTC), 255, 266
  - asymptotic performance of, 264
  - concept of, 264
- Multiple description vector quantizers (MDVQ), 264, 280
  - based image watermarking scheme, 284
  - labeling problem, role in, 264
  - robust audio communication, 267
- Multiple state technique, 284
- Multi-point relays (MPRs), 349
- Multi-resolution coding (MRC), 283
- Multi Scale-Structural SIMilarity (MS-SSIM), 179
- Multiservice business router (MSBR), 268
- Multi-touch technologies, 325
- Multi-tree construction schemes, 351
- Multi-view Video plus Depth (MVD) format, 140

**N**

- National Institute for Research in Computer Science and Control (INRIA), 418  
 Near-threshold strategy, 241–242  
 Netravali-Robbins method, 39  
 Network coding (NC)  
     butterfly network problem, 364  
     concept of, 364  
     data dissemination, 367–368  
     Linear network coding, 365  
     Practical Network Coding, 366  
 Network Coding for Video (NCV) technique, 369  
 Next generation video compression technology. *See also* High Efficiency Video Coding (HEVC)  
     HD resolutions, 97  
     screen content, 99  
     Ultra high-definition TV (UHDTV), 98  
 No false alarm (NoFA) profiles, 444–446  
 Noise-to-mask ratio (NMR), 266  
 Non-rigid textures, 213–215  
 Non-square Quad Tree (NSQT), 188  
 No-reference methods, 240  
 Normalized detection cost rate (NDCR), 444  
 Normalized interpolation Error (NE), 84  
 Novel line spectral pairs (LSP)-based MDC method, 266  
 NTP (Network Time Protocol), 334  
 Nyquist sampling theorem, 265

**O**

- Objective quality evaluation, 107  
     coding with reduced block sizes, 110–113  
     HEVC vs. H.264/AVC, 107–109  
     TS for screen content, 112–114  
 Objective quality metrics, 228  
     classification, 240  
     perception oriented image and video quality metrics  
         based on statistical models, 243  
         HVS-based quality metrics, 242  
         quality feature-based, 243  
         transform-based, 244  
     performance comparison, 244  
     primary uses of, 239  
     PSNR, characterization of, 240–241  
 On-demand streaming, 329  
 OpenCV (OpenCV 2013), 429  
 Optical flow, 28, 32–33  
     definition, 29  
     equation, 31  
     quantitative assessment, 83  
     vs. 2D motion vector field, 29

- Optical flow estimation approaches  
     dense motion vector field, 38  
     disadvantages, 38  
     Horn-Schunck algorithm, 37  
     Lucas-Kanade method, 37  
     on-line databases, 39  
 Optimized Link-State Routing (OLSR), 349  
 Oriented BRIEF (ORB), 428  
 Outer boundary matching algorithm (OBMA), 304  
 Outlier ratio (OR), 239  
 Overlapped block motion compensation (OBMC), 78–79, 305, 308  
 Oxford Building dataset, 383

**P**

- Packing method, 386  
 Pairwise correlating transform (PCT), 255, 261  
 Panum's fusional area, 121  
 Parallax, 122  
 Parametric motion estimation  
     direct, 67  
     indirect, 67  
 Partial differential equation (PDE), 180–181, 305  
     based approaches, 205  
         intra prediction based on Laplace PDE, 205  
         intra prediction based on TV, 206  
         optimization strategies and performance, 206–207  
 PCA-SIFT, 397, 428  
 PCR (Program Clock Reference), 334  
 Peak-Signal-to-Noise-Ratio (PSNR), 84–85, 107, 177, 189  
     average performance, 302, 309–310, 312  
     characterization of, 240  
 Peak-Signal-to-Perceptible-Noise-Ratio (PSPNR), 177  
 Pearson Linear Correlation Coefficient (LCC), 238  
 Peking University (PKU), 418  
 Perception-based Video quality Metric (PVM), 243  
 Perceptual audio coder (PAC), 266  
 Perceptual Distortion Metric (PDM), 178  
 Perceptual quality degradationD video  
     during 3D content creation process, 147  
     display device, distortion by, 148  
     view synthesis distortion, 148  
 Perceptual video codecs (PVCs)  
     image analysis  
         block-based schemes, 181  
         IAC approach, 180  
         region-based IAC-VC schemes, 181–182  
     image completion  
         non-parametric, 181  
         parametric, 180  
     Partial Differential Equation based, 180–181

- integration of JND and low complexity, 182
- IDSQ performance assessment, 187
- low computational complexity implementation, 183, 185
- picture characteristics, flexibility in, 183, 186
- JND, integration of, 177
  - with adaptive masking slope, 178
  - 3D pixel domain subband PVC, 177
  - fine perceptual rate allocation, 179
  - foveated pixel domain model, 179
  - perceptual pre-processing of motion compensated residuals, 177
  - transform coefficients, perceptual suppression of, 178–179
- Perceptual video coding tools
  - coding stages
    - in-loop filter, 160
    - quantization, 160
    - rate-distortion optimization, 160
  - design of, 160, 167
- Perspective/projective motion model, 35
- Phase-correlation methods, 43
- Picture-in-picture (PiP) detection, 422–423
- Picture partitions
  - coding tree unit (CTU), 101
  - coding unit structure, 101–102
  - prediction unit structure, 101–102
  - transform unit structure, 103
- Picture slices, 96
- Picture types, 95
- Pixel/pel-recursive approaches
  - advantages of, 42
  - Cafforio-Rocca algorithm, 40
  - DFD, definition of, 39
  - Netravali-Robbins method, 39
  - remarks on, 42
- Platelet coding, 145
- Point-of-care technology, 4
- Polynomial model, 33–34
- Post-processing techniques
  - geometric verification by geometric coding
    - affine estimation for enhancement, 410–411
    - with coding maps, 409, 411
    - fan coding, 407
    - ring coding, 408
  - query expansion, 406
- Predicted (P) picture, 95
- Predicted (P) frames, 134
- Prediction blocks (PBs), 76
- Prediction unit (PU), 101–102
- Predictive (P) frames, 332–333
- Priority Encoding Transmission (PET), 284
- Projection onto convex sets (POCS), 355
- Proprioception, 120
- P (Predicted) slice, 96
- Q**
- Quadratic motion model, 35
- Quality assessment, 216
  - challenges and alternatives measures, 219
  - global spatial, 216
  - global temporal, 217
  - local spatial, 218
  - local temporal, 218–219
  - for synthesized textures, 216
- Quality Assessor (QA), 209
- Quality layer integrity check signaling, 355
- Quality of Experience (QoE), 148, 268
- Quality of Service (QoS), 268, 356
- Quantization
  - coefficient, 11, 13
  - feature (*see* Feature quantization)
  - perceptually optimized video compression, 160
  - process
    - integration of JND model in, 173
    - tools, 175
    - Transform Unit, 103
  - Quantization matrices, 175
  - Quantization noise, 132
  - Quantization Parameters (QP), 79, 106, 184
  - Quantized frame expansion (QFE), 362
- Quarter Common Intermediate Format (QCIF), 95, 97
- R**
- Random Access (RA), 106
- Random assessment delay (RAD), 349
- Random Linear Network Coding (RLNC), 253, 366
- RANdom Sample Consensus (RANSAC)
  - verification, 433–434
- Rate-distortion optimization (RDO), 76, 157, 352, 369
- Rate Distortion (or Quality) Optimization RDO (RQO), 228
- Rate-less codes, 351
- Rate-quality optimization, 239, 245
- Rate *vs.* quality, 5
- Real-Time Control Protocol (RTCP), 336
- Real-time transport protocol (RTP), 333–334
- Real wavelets, 46
- Reduced reference methods, 240
- Redundancy rate-distortion (RRD) analysis, 256, 260
- Redundancy *vs.* error resilience, 5
- Redundant picture (RP) tool, 353
- Reference, 47

Region-based video coding techniques  
 framework, 209  
 quality assessment  
   challenges and alternatives measures, 219  
   global spatial, 216  
   global temporal, 217  
   local spatial, 218  
   local temporal, 218–219  
   for synthesized textures, 216  
 system integration and performance, 219–220  
   rate-quality performance, 221  
   system configuration, 220  
 texture analysis, 210–211  
 texture synthesis  
   non-rigid textures, 213–215  
   rigid textures, 212–213  
 Regression analysis, 238  
 Residual QuadTree (RQT), 103  
 R-estimators, 69  
 Reversible variable length coding (RVLC), 354  
 Robust audio tool (RAT), 266  
 Robust entropy coding (REC) methods, 354  
 Robust wave-form coding (RWC), 353  
 Root Mean Squared Error (RMSE), 239  
 Run-length coding (RLC), 14

## S

Scalable cascaded hashing (SCH) approach, 396  
 cascaded hashing, 400, 397  
 dimension reduction on SIFT by PCA, 397–399  
 matching verification, 401  
 Scalable video coding (SVC)  
   quality scalability, 357  
   spatial scalability, 357  
   temporal scalability, 356  
 Scale Invariant Feature Transform (SIFT)  
   descriptor, 428  
   detector, 427–428  
 Screen content (SC), 94–95, 99  
   transform skipping for, 112–114  
 Search window, 52  
   conjugate directions search, 51  
   diamond search, 54–56  
   2D-logarithmic search, 51, 53  
   fast search, 50  
   full search, 49  
   hexagon search, 55–56, 58  
   three-step search, 52, 54  
 Semantic indexing, 418  
 Semi-supervised hashing method (SSH), 394  
 Set partitioning in hierarchical trees (SPIHT), 284

Shazam (audio domain), 417  
 Short-time Fourier transform (STFT), 438  
 Shot boundary detection (SBD), 418  
   aim of, 419  
   components in, 421  
   intra-frame and inter-frame features, 421  
 TRECVID, 420  
   evaluation criteria, 442  
   evaluation results, 443  
   video segmentation and extract keyframes, 420  
 Significant coefficient decomposition (SCD), 271  
 Single Description Coding (SDC), 359  
 Single stimulus methods, 232  
 Single Stimulus Continuous Quality Evaluation  
   method, 232  
 Singular Value Decomposition (SVD), 68  
 Small diamond-shape-partner (SDSP), 54–55  
 Smartphones, 329  
 Smooth Pursuit Eye Movement (SPEM), 167  
 SnapTell (image domain), 417  
 Spatial activity (SA), 312  
   measurement of, 313–314  
 Spatial error concealment (SEC) method  
   advantages of, 298  
   ingredients, 297  
   method  
     boundary pixel selection, 297  
     coarseness measure, 299  
     interpolation of lost pixels, 297–299  
     performance of, 302–303  
     required techniques, 300  
       bilinear interpolation, 301  
       directional entropy, 301  
       directional interpolation, 300  
       variations, 300  
 Spatial information (TI) measures, 230  
 Spatially scaled method, 284–285  
 Spatial redundancy, 330  
 Spatio-temporal MAD (ST-MAD), 242  
 Spearman Rank Order Correlation Coefficient  
   (SROCC), 238  
 Spectral hashing method, 394  
 Speeded Up Robust Features (SURF), 428  
 Standard Definition Television (SDTV), 95  
 Stereo disparity, 122  
 Stereo photography, 129–130  
 Stereoscopic and multi-view video coding  
   applications of  
     3D cinema, 129  
     3D television, 130  
   compression of  
     depth-based 3D video formats, 140–141

- depth estimation (*see* Depth synthesis)
  - depth map coding, 144
  - joint coding of texture and depth, 146
  - using inter-view prediction, 136–137
  - video coding, 131
  - virtual synthesis using DIBR, 138
  - 3D display technologies, 123
    - autostereoscopic displays (*see* Autostereoscopic displays)
    - stereoscopic displays, 123
  - development of, 120
  - fundamentals of
    - monoscopic depth cues, 122
    - proprioceptive depth cues, 120–121
    - stereoscopic depth cues, 122
  - quality evaluation of 3D video, 147
    - perceptual quality degradation in, 147
    - standards and metrics for, 148
  - Stream switching, 337
  - Structural Similarity Image Metric (SSIM), 243
  - Structural SIMilarity (SSIM) metric, 158, 219
  - Structured dual vector quantizers (SDVQ), 267
  - Subjective Assessment Methodology for Video Quality (SAMVIQ), 232
  - Subjective quality assessment, 228
    - methodology for, 229
    - statistical analysis of
      - confidence interval, 234
      - MOS and DMOS, 233
      - screening of observers, 234
    - subject selection, 231
    - test conditions, 231
    - testing environment, 231
    - testing methodology, 231
      - double stimulus methods, 233
      - single stimulus methods, 232
      - triple stimulus methods, 233
    - test material, 230
  - Subjective quality assessment, 85, 228, 246, 277
  - Subjective quality evaluation, 106
  - Subjective video databases
    - EPFL-PoliMI, 237
    - evaluating metrics using, 237
      - correlation analysis, 238
      - regression analysis using logistic function, 238
    - IRCCyN/IVC HD, 237
    - IVP, 237
    - LIVE, 237
    - NYU subjective dataset I–IV, 237
    - primary uses of, 235
    - VQEG FRTV, 237
    - VQEG HDTV, 237
    - VQEG Multimedia Phase I database, 237
  - Sum of absolute differences (SAD), 57, 304
  - Sum of Squared Differences (SSD), 57
  - Super Hi-Vision (SHV), 155
  - Super multi-view systems, 127–128
  - Supporting visual words, 395
  - Support vector machines (SVM), 421
  - Supra-threshold
    - condition, 174, 191
    - HVS strategy, 241–242
  - Switching I (SI), 96
  - Switching P (SP), 96
  - Symbol encoding
    - entropy encoding, 14
    - sparse matrices, 14
  - Synchronization codeword (SC), 354
- T**
- Template matching
    - algorithm, 207
    - averaging, 207–208
    - priority-based, 208
    - technique, 66, 70
  - Temporal Contrast Sensitivity Function (CSF), 167
  - Temporal error concealment (TEC) method
    - ingredients, 302
    - method, 303
      - matching measures, 304
      - motion vector refinement, 305
      - selecting motion vector candidates, 304
    - performance of, 308–312
    - required techniques, 306
      - boundary matching error, 306–307
      - external boundary matching error, 306–307
      - overlapped block motion compensation, 308
      - variations, 306
  - Temporal information (TI) measures, 230
  - Temporal level zero index signaling, 355
  - Temporally-Ordered Routing Algorithm (TORA), 347
  - Temporal redundancy, 330
  - Term frequency inverse document frequency (TFIDF), 431
  - Test Model 5 (TM5), 175
  - Texture analysis and synthesis (TAS) for video compression
    - block-based video coding techniques
      - framework, 199–201
      - non-parametric approaches, 206
      - parametric approaches, 201
      - PDE-based approaches, 205
      - perception-oriented, 201
    - High Efficiency Video Coding, 198
    - international standards, 198
    - multimedia communication and storage, 198

- perception-oriented video coding, 199  
 region-based  
     framework, 209  
     quality assessment, 216  
     system integration and performance, 219  
     texture analysis, 210  
     texture synthesis, 212  
 Texture Analyzer (TA), 209  
 Texture segmentation, 212  
 Texture Synthesizer (TS), 209  
*The Bell* (1980), 255  
 Three-step search (TSS) algorithm, 52  
 Tineye (search engine), 384  
 Total Variation (TV) methods, 38  
 Total variation (TV) model, 205  
 Transform Skip (TS), 188  
 Transform unit (TU) structure, 103  
 Translational motion model, 34  
 TRECVID  
     content-based copy detection results  
         evaluation results, 444–447  
         task and evaluation criteria, 443  
     instance search task (INS)  
         enhancements for, 449  
         evaluation criteria, 449  
         evaluation results, 449–450  
     shot boundary detection results  
         evaluation criteria, 442  
         evaluation results, 443  
 Triple stimulus methods, 233  
 Triple stimulus continuous evaluation scale (TSCES)  
     method, 220, 233
- U**
- UKBench dataset, 383  
 Ultra High-Definition Television (UHDTV), 7, 94, 155  
 Unequal error protection (UEP), 351, 353, 368  
 Universal Image Quality Index (UIQI), 243
- V**
- Variable-Length Coding (VLC), 95–96, 331  
 Variable-size block matching techniques, 63, 66  
 Vector quantizer (VQ), 267  
 Velocity vector field, 32–33  
 Vergence, 120–122  
 VidCat service, 448  
 Video coding  
     basics of  
         architecture, 8  
         coding tree units, 9  
     macroblocks, 9–10  
     quality assessment, 9  
     still image encoding, 8  
     video encoding, 8  
 compression of stereoscopic and multi-view video (*see also*  
     Stereoscopic and multi-view video coding)  
     correlation and entropy reduction, 132  
     encoder optimization, 134  
     entropy coding, 132  
     information, 131  
     inter-frame prediction, 133  
     decorrelating transforms  
         coefficient quantization, 11, 13  
         discrete cosine transform, 11–12  
         mechanisms, 10  
     HVS sensitivity modeling, 162  
         contrast masking, 165  
         frequency masking, 162  
         JND, evaluation of, 170, 174  
         luminance masking, 164, 171  
         in spatio-temporal, combination of masking, 170  
         temporal masking, 167  
     JND model, integration of  
         in-loop filter process, 175  
         in quantization process, 173, 185  
         rate-distortion optimization process, 175  
         standard functionalities to support, 175  
 motion estimation  
     block-based motion-compensated video coding  
         architecture (*see* Block-based motion-compensated  
             video coding architecture)  
     FD and DFD frames, pdf of pixel values of, 15–16  
     inter-frame redundancy, 15  
     motion compensated prediction, 15  
     perceptual coding tools (*see* Perceptual video coding tools)  
     perceptual video coding schemes (*see* Perceptual video  
         codecs (PVCs))  
     requirements  
         bandwidth availability, 7  
         bit rate requirements, 6  
         desirable features, 5  
         ratio, 6  
         trade-offs, 5  
     standardization  
         bitstream format and decoding process, 20  
         chronology of, 20–21  
         HEVC, 20  
         history of, 20  
         interoperability, 20  
         performance of, 22  
         scope of, 20  
     symbol encoding, 14

- entropy encoding, 14
- sparse matrices, 14
- video technology, 3
  - business and automation, 4
  - consumer video, 4
  - healthcare, 4
  - surveillance, 4
- Video compression
  - next generation formats and content
    - screen content, 99
    - ultra high-definition TV, 98
  - new market requirements, 99
  - standard, history of
    - H.261, 95
    - H.263, 95
    - H.264/AVC (*see* H.264/AVC)
    - H.262/MPEG-2 Video (MPEG-2), 95
    - MPEG-4, 96
    - picture types, 95
    - previous standards, development of, 95
    - technical characteristics of, 96
  - texture analysis and synthesis (*see* Texture analysis and synthesis (TAS) for video compression)
- Video copy detection, 326
  - applications, 417
  - audio-based, 418
    - acoustic features, 437
    - advantage of, 436
    - audio fingerprint, 438
    - audio normalization, 437
    - block diagram of, 437
    - index and search, 439
    - module, 437
  - benefits, 418
  - copy and near-duplicate detection
    - personal media organization, 448–449
    - TRECVID (*see* TRECVID)
  - goal of, 418
  - joint audio- and visual-based
    - audio and visual fusion schemes, 440
    - multi-query result normalization and fusion, 442
  - region and partial content search
    - mobile content-based copy detection, 450–451
    - TRECVID INS (*see* TRECVID)
  - visual-based, 418
    - block diagram of, 419–420
    - keyframe matching based on LSH Hash/BoW method, 433
    - query keyframes, detection of, 419–420
    - RANSAC verification, 433–434
    - reference/query normalization, 434–435
    - SBD algorithm (*see* Shot boundary detection (SBD) algorithm)
- score normalization, 436
- transformation detection and normalization, 422
- video level result fusion, 435–436
- visual feature (*see* Visual feature)
- Video error concealment methods, 295
  - mode selection
    - ingredients, 309
    - MCTA and SA, measurement of, 313–314
    - method, 310, 313
    - performance of, 314
    - variations, 312
  - multi-mode approaches, 315–316
    - performance of, 315–317
  - spatial error concealment
    - ingredients, 297
    - method, 297
    - performance of, 302–303
    - required techniques, 300
    - variations, 300
  - temporal error concealment
    - ingredients, 302
    - method, 303
    - performance of, 308–312
    - required techniques, 306
    - variations, 306
- Video Object Plan (VOP), 81
- Video Quality Experts Group (VQEG) FRTV Phase I programme, 237
- Video quality measurement
  - approaches to, 228
  - goal of, 228
  - influential factors, 229
  - objective quality metrics
    - classification, 240
    - perception oriented, 242
    - performance comparison, 244
    - primary uses of, 239
    - PSNR, characterization of, 240–241
  - subjective
    - datasets (*see* Subjective video databases)
    - testing (*see* Subjective quality assessment)
- Video Quality Metric (VQM), 179
- View interpolation, 140
- Vision-based systems, 4
- Visual communications, 4
- Visual feature, 424
  - extraction
    - global, 424
    - local, 427
  - frame level result fusion, 434
  - indexing and search, 429
    - bag of visual words method, 431–432

*k-d* tree method, 429  
locality sensitive hashing, 430–431  
Visual Signal-to-Noise Ratio (VSNR), 242  
VLSI semiconductor technologies, 325  
Vocabulary tree method, 432

subsampling problem, 43  
Wheatstone, Charles, 129  
Wireless channel  
    characteristic of, 253  
    multimedia traffic over, 253  
Wyner-Ziv Coding (WZC), 364

## W

Wavelet domain  
    motion estimation in  
        optical flow constraint, 46  
        subsampled subbands, 44, 48

## Z

Zero-mean Normalized SSD (ZN-SSD), 59  
Zettabytes, 3  
Zig-zag scanning pattern, 14