Lecture Notes in Computer Science 1035
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Advisory Board: W. Brauer D. Gries J. Stoer

S.Z. Li D.P. Mital E.K. Teoh H. Wan (Eds.)

Recent Developments in Computer Vision

Second Asian Conference on Computer Vision,
ACCV '95
Singapore, December 5-8, 1995
Invited Session Papers

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany

Juris Hartmanis, Cornell University, NY, USA

Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Stan Z. Li
Dinesh P. Mital
Eam Khwang Teoh
Han Wang
Nanyang Technological University
School of Electrical and Electronic Engineering
Nanyang Avenue, Singapore 639 798, Singapore

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Recent developments in computer vision : invited session papers / Second Asian Conference on Computer Vision, ACCV '95, Singapore, December 5 - 8, 1995. S. Li ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1996
(Lecture notes in computer science ; 1035)
ISBN 3-540-60793-5
NE: Li, Stan Z. [Hrsg.]; Asian Conference on Computer Vision <2, 1995, Singapore>; GT

CR Subject Classification (1991): I.3, I.5, I.2.9-10, I.4

Cover Illustration taken from the contribution by Papachristou, Petrou, and Kittler on page 392.

ISBN 3-540-60793-5 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

© Springer-Verlag Berlin Heidelberg 1996
Printed in Germany

Typesetting: Camera-ready by author
SPIN 10512499 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Foreword

Collected in this volume are 56 invited session papers presented at the Second Asia Conference on Computer Vision (ACCV) held in Singapore, 5–8 December 1995. The theme of the conference was "Computer Vision for the Twenty First Century". Fifteen leading researchers, whose names are listed below, were invited as session organizers to solicit and review these papers. The papers cover various areas in computer vision and address both theoretical and practical issues. In my view, they reflect the state-of-the-art developments in computer vision.

As the invited sessions' chair, I would like to thank the organizers and paper contributors for their efforts in making the ACCV'95 invited sessions a successful event. Many thanks also to the team of the conference organizing committee for their excellent work in running the conference.

December 1995 S.Z. Li

Invited Session Organizers

J. Aggarwal	University of Texas, TX
K. Baker	University of Reading, UK
P. Bouthemy	IRISA/INRIA, France
T.S. Huang	University of Illinois, IL
Y.P. Hung	Academia Sinica, Taipei
A.K. Jain	Michigan State University, MI
J. Kittler	University of Surrey, UK
S. Lakshmanan	University of Michigan, MI
S.D. Ma	Chinese Academy of Sciences, Beijing
K.V. Mardia	University of Leeds, UK
M. Mohiuddin	IBM Almaden Research Center, CA
R.W. Picard	MIT, Cambridge, MA
J. Thijssen	University Hospital, Netherlands
S. Ullman	Weizmann Institute of Science, Israel
A. Zisserman	Oxford University, UK

Contents

New Directions
Chair: *J. Aggarwal*

Digital Libraries: Meeting Place for Low-Level And High-Level Vision 3
R. Picard

Computer Vision in Next Generation Image and Video Coding 13
T.S. Huang, R. Lopez

Video Annotation: Computers Watching Video 23
A. Bobick

Video Representations 33
R. Bolle, Y. Aloimonos, C. Fermüler

Analysis of Left Ventricular Motion 43
W.-H. Liao, S.J. Aggarwal, J.K. Aggarwal

Applications of Computer Vision in Space Robotics 59
C.-H. Chien

Motion Estimation and Segmentation
Chair: *P. Bouthemy*

Optical-Flow Estimation while Preserving Its Discontinuities: A Variational Approach 71
R. Deriche, P. Kornprobst, G. Aubert

Space- and Time-Variant Estimation Approaches and the Segmentation of the Resulting Optical Flow Fields 81
H.-H. Nagel, A. Gehrke, M. Haag, M. Otte

Combining the Hough Transform and Multiresolution MRF's for the Robust Motion Estimation 91
M. Bober, J. Kittler

Direct Model-Based Image Motion Segmentation for Dynamic Scene Analysis 101
J.-M. Odobez, P. Bouthemy

Structure from Motion
Chair: *T.S. Huang*

Modeling Structured Environments Using Robot Vision 113
S. Shah, J.K. Aggarwal

Design Trade-Offs for Real-Time Block Matching Motion Estimation Algorithms 129
Z. He, M.L. Liou

Pixel Matching and Motion Segmentation in Image Sequences 139
N. Ahuja, R. Charan

On the Motion Problem of a Rigid Planar Patch and its Invariance 149
C.-H. Lee

Virtual-Space Teleconferencing – Real-Time Detection and Reproduction of 3D Face and Body Images 159
F. Kishino, K. Ebihara, J. Ohya

Active Vision

Chair: *Y.P. Hung*

High-Performance Tracking with TRICLOPS 171
A. Wavering, H. Schneiderman, J.C. Fiala

Developing an Active Observer 181
J.-O. Eklundh, T. Uhlin, P. Nordlund, A. Maki

Active Stereo Vision System with Foveated Wide Angle Lenses 191
Y. Kuniyoshi, N. Kita, S. Rougeaux, T. Suehiro

A Control Theoretical Approach to Active Vision 201
H.I. Christensen, J. Horstmann, T. Rasmussen

Active Rangefinding and Recognition with Cubicscope 211
Y. Sato

Real-Time Vision Is Not Frame-Rate Image Analysis 219
L. Kitchen, T.-K. Cheng, Z.-Q. Liu

Stereo Vision

Chair: *S.D. Ma*

Active Vision Based Stereo Vision 231
C. Wang, Z. Hu, S.D. Ma

Motion of a Stereo Rig: Strong Weak and Self Calibration 241
Z. Zhang

Color Stereo Vision: Use of Appearance Constraint and Epipolar Geometry for Feature Matching 255
M. Xie, L.Y. Liu

Unifying Stereo, Motion and Object Recognition via Epipolar Geometry 265
G. Xu

Constructing a Visual Memory in Terms of Uniqueness 275
J.-Y. Zheng

Statistics and Image Analysis

Chair: *K.V. Mardia*

Statistical Models and Algorithms for Tomography 289
M. Hudson

Shape Statistics and Image Analysis 297
K.V. Mardia

Cluster Monte Carlo Algorithms and Their Applications 307
J.-S. Wang

Image Estimation

Chair: *S. Lakshmanan*

Gibbs Sampler and Maximum Likelihood Estimation for Unsupervised Image Segmentations 319
C.S. Won

Lane Boundary Detection Using Deformable Templates: Effects of Image Subsampling on Detected Lane Edge 329
K. Kluge, S. Lakshmanan

Contextual Pattern Interpretation

Chair: *J. Kittler*

Clique Finding Relaxation Labeling Networks 343
M. Pelillo

Rectifying Structural Matching Errors 353
E.R. Hancock, R.C. Wilson

Parallel Distributed Relaxation Labeling 363
S.Z. Li

Subgraph Isomorphism Detection in Polynomial Time on Preprocessed Model Graphs 373
B.T. Messmer, H. Bunke

Line Postprocessing Using Probabilistic Relaxation and Relative Measurements 383
P. Papachristou, M. Petrou, J. Kittler

Geometric Invariance

Chair: *A. Zisserman*

Multiple-View Geometry and Photometry 395
A. Shashua

Euclidean 3D Reconstruction from Stereo Sequences with Variable Focal Lengths 405
M. Pollefeys, L. Van Gool, T. Moons

Characterizing the Stability of 3D Invariants Derived from 3D Translational Symmetry 415
J. Liu, E. Walker, J. Mundy

MORSE: An Architecture for 3D Object Recognition Based on Invariants 425
J. Mundy, R. Curwen, J. Liu, C. Rothwell, A. Zisserman, D. Forsyth

Object Recognition

Chair: *S. Ullman*

Finding Human Faces with a Gaussian Mixture Distribution-Based Face Model 437
T. Poggio, K.-K. Sung

Handwriting Recognition 447
T.M. Breuel

An Integrated Model for Evaluating the Amount of Data Required for Reliable Recognition 457
M. Lindenbaum

3D Object Recognition Using Bidirectional Modular Networks 467
H. Ando

Outdoor Computer Vision

Chair: *K. Baker*

A Region Extension Method for Natural Outdoor Scene Interpretation 479
M. Hild, Y. Shirai

Image Processing Techniques for Crowd Density Estimation Using a Reference Image 489
J.H. Yin, S.A. Velastin, A.C. Davies

Autonomous Road Vehicle Guidance in Normal Traffic 499
F. Thomanek, E.D. Dickmanns

Evaluation of Image Sequences from Outdoor Scenes: Selected Problems and Solutions 509
H.-H. Nagel, H. Kollnig

Vehicle Localisation and Discrimination in Outdoor Traffic Scene 519
T.N. Tan, G.D. Sullivan, K. Baker

Content-Based Retrieval

Chair: *R.W. Picard*

SWIM: A Prototype Environment for Visual Media Retrieval 531
H.-J. Zhang

Video Segment Indexing through Classification and Interactive View-Based Query 541
J.C.M. Lee, W. Xiong, D.G. Shen, R.H. Ma

View-Based Techniques for Searching for Objects and Textures 551
M.J. Swain, C.H. Frankel, M. Lu

Graph Grammar Based Object Recognition for Image Retrieval 561
C. Klauck

Similarity Matching 571
S. Santini, R. Jain

Document Image Analysis

Chair: *M. Mohiuddin*

A Document Image Analysis and Recognition System for Japanese Family Registration 583
T. Amano, K. Toyokawa, T. Mano, S. Toriyama

Applying Handwriting Recognition to US Census Forms 593
T.M. Breuel

Author Index 603

New Directions

Digital Libraries: Meeting Place for High-Level and Low-Level Vision

Rosalind W. Picard

MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 USA
picard@media.mit.edu
http://www.media.mit.edu/~picard/

Abstract. The average person with a networked computer can now understand why computers should have vision – to search the world's collections of digital video and images and "retrieve a picture of ___." Computer vision for intelligent browsing, querying, and retrieval of imagery is needed now, and yet traditional approaches to computer vision remain far from a general solution to the scene understanding problem. In this paper I discuss the need for a solution based on combining high-level and low-level vision, that works in concert with input from a human user. The solution is based on: 1) Learning from the user what is important visually, and 2) Learning associations between text descriptions and visual data. I describe some recent results in these areas, and overview key challenges for future research in computer vision for digital libraries.

1 Introduction

Collections of digital imagery are growing at a rapid pace. The contexts are broad, including areas such as entertainment (e.g. searching for an actress washing her hair), education (e.g. finding illustrations of pilgrims), science (e.g. analyzing satellite imagery), medicine (e.g. comparing onset rates of osteoporosis), marketing (e.g. insuring the competitor hasn't used leopards in their ads), and design (e.g. selecting an oriental rug). In all these applications and more, vision tools can facilitate access to content.

1.1 Vision, signal processing, and common sense

Computer vision is not a solo solution to the problems of retrieval and annotation in image and video libraries. There are numerous signal processing issues such as compression, and analysis of accompanying non-visual signals such as the soundtrack. There are also business issues (e.g. billing for downloading data), database issues (e.g. fast algorithms for indexing), interface issues (e.g. visualization and manipulation of multiple video streams), and numerous other research problems. Natural language queries and common sense systems also play a significant role; for example, Lenat's CYC common sense reasoning system can take a request such as "find someone wet," and return an image with a label such as "man

finishing a marathon" [1]. The computer vision solutions will work best if wisely integrated with solutions from these other domains.

In [2] I overviewed the latest digital library research issues for the image processing community to address. The emphasis for that community is on finding models for simultaneous compression and content description, and improving measures of visual similarity for comparing images. However, the domains of the image processing and computer vision communities overlap, so it is wise for them to watch what each other does to insure complementary efforts and avoid wasteful duplication of errors, especially in spatiotemporal segmentation and modeling where research seems to overlap the most. In this paper I will focus on important research problems that I did not discuss in [2], problems which are traditionally closer to the computer vision community – namely, generating descriptions of image content, and vision systems that learn.

Although my focus in the rest of this paper is on computer vision for digital libraries, much of the following discussion also applies to other perceptual domains such as acoustic scene analysis. Furthermore, construction of vision tools (such as a visual thesaurus) should not proceed in isolation, but in concert with the other perceptual domains (e.g. audio thesaurus and text thesaurus), since humans, the ultimate judges of these systems, make associations which weave through all the perceptual senses.

1.2 The importance of bias in learning

Saying someone is "biased" is usually not a compliment; however, bias ultimately is what leads someone to a conclusion, right or wrong. Bias can not only be favorable, but it can also be essential for good performance, especially when the number of possibilities is large. The word "bias," as used in the rest of this paper, refers to that which can be controlled, and which guides a system to its answer.

Consider a system where all solutions are equally likely, and a solution is pursued by an optimization algorithm. If the algorithm falls into local minima, or can be varied to provide different equally optimal solutions, then the algorithm itself is a type of *procedural* bias. If the space of the optimization contains regions with different likelihoods, then these likelihoods are a form of *declarative* bias. In fact, the initial constraints established to set up the problem can be considered a form of declarative bias, ad infinitum. Moreover, the distinction between declarative and procedural is not firm; a procedure's text is also a "declaration." Bias can be anything – priors, weights, procedures, criteria, the space of possibilities permitted – which the designer gives the system for use in choosing its answers.

In theory, each 2-D image has an infinity of 3-D images which could have given rise to it. In visual recognition and retrieval where the space of possibilities is astronomical, a system will effectively never return from its search for a solution unless it has some bias.

In fact, there is recent neurological evidence supporting the role of emotions as a critical biasing mechanism in human decision-making [3]. The evidence indicates that humans with a particular kind of brain damage do not have these emotional biasing mechanisms, and consequently suffer a loss of decision-making

ability. When asked, say, to find a good time to schedule an appointment, they disappear into an endless space of possibilities and cannot reach an answer on their own.

2 Combining low- and high-level vision with learning

Traditional low-level vision consists of the underlying representations used for the image and video – edges, color, texture, etc. The stored feature values comprise a low-level declarative bias, which is used upstream to make decisions.

Higher-level vision is usually concerned with interpretations of low-level features. Consider the selection of a recognition procedure that is well-suited to labeling one of the categories in a scene. The available procedures, and the mechanism(s) for their selection comprise what may be considered a high-level bias.

Successful biases must happen at both the low and high levels, and depend both on the data and the goals. With retrieval systems, where a user is in the loop, it is impossible to specify all the goals in advance. A key challenge for vision researchers is to develop a system with a good bias, that also knows how to intelligently change this bias as the goals and data change.

The strategy outlined in this paper addresses the problem of developing such a learning vision system, that learns its biases continuously. It provides a powerful solution for digital libraries and other domains where the following two Learning Criteria are met:

1. The system can pre-compute low-level biases and can analyze its own learning performance. These may be performed as *offline learning.*
2. The user of the system is present, and interacts with the system. Ideally, this allows *online learning.*

2.1 Teaching the system to learn

In traditional pattern recognition, the designer of the system runs extensive experiments comparing different features of the data and different recognition strategies, until finally the best combination of features and decision-making algorithms is found. Usually this process is conducted for a specific type of data, and a specific set of goals.

In digital libraries, there is not time for the researcher to follow this traditional paradigm for *every* new set of data, for *every* set of recognition goals. In other words, it is now necessary to automate more of the iterative learning process usually done by the researcher.

Minka and Picard have built a system, "FourEyes," which satisfies the two Learning Criteria above. Although I refer the reader to [4] for details of how it learns, how it compares to other learning systems and how its performance has been evaluated, I will highlight a few of its features below to illustrate the

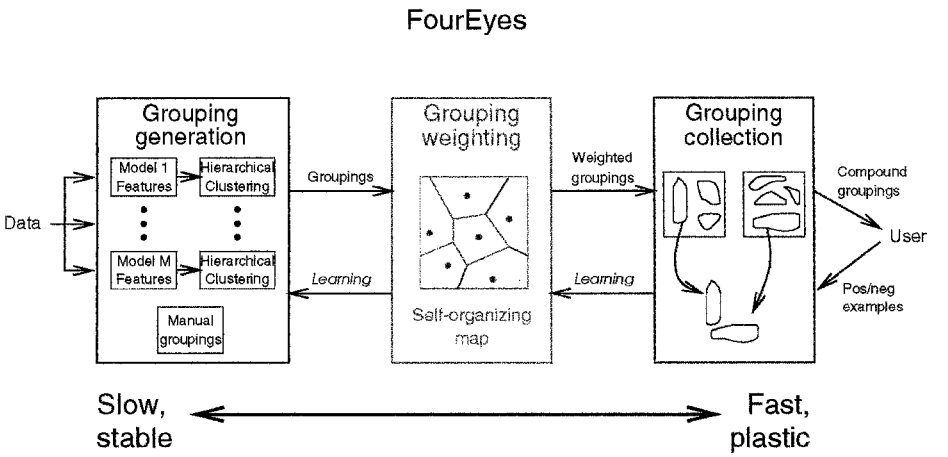


Fig. 1. The FourEyes learning system. The left-most box computes *groupings* for all the data in the database, learning offline. The right-most box interacts at run-time with the user, determining which groupings or combination of groupings best represent the data that interests the user, learning online. The middle box learns about learning, so to speak. It replaces a bias which would typically be procedural with one which is declarative, and therefore easier to update for changing the longer-term behavior of the system. For example, if the system sees the same problem repeatedly, then the middle box enables it to become faster at solving that problem, and problems similar to that problem. The middle box provides a way for a learning system to select and modify its own bias.

arguments in this paper. A diagram of the FourEyes learning system is given in Fig. 1.

FourEyes was originally constructed to assist the user in annotating, or attaching text descriptions, to databases of imagery. Because it is tedious for a user to label every region in every image, annotation is a good task for machine vision. However, because people segment and label regions differently depending on a variety of factors, it is not a task that has a single optimal solution. Instead, a system is needed that learns how to classify (annotate) regions the way that a particular user is annotating them.

2.2 Too much bias...too little bias...

Traditional segmentation and classification systems rely on a single model and set of decision-making rules. The results, such as for segmentation on satellite imagery, work well under carefully controlled conditions for the categories the system is trained to recognize, but can give wildly incorrect solutions when the controls are violated, such as when data outside those categories is shown to the system (an admittedly unfair test for the traditional purpose, but one which

arises in the new diverse world of digital libraries). Such systems may be said to be over-biased.

Most of the latest segmentation and labeling systems try to improve flexibility by using one powerful model such as a Markov random field for the segmentation, and then allowing for model parameters to vary within regions, and for labels to be associated with particular parameter ranges. Posed probabilistically, these systems tend to be doubly-stochastic with astronomical computational demands for their simultaneous parameter estimation and segmentation. Because they allow too large a space of solutions, they rarely find optimal solutions, and rarely converge except for trivial problems. In short, they do not have enough bias.

In contrast with traditional approaches, the FourEyes system does not have a fixed bias. Instead, it is able to select and modify its bias, taking on some of the iterative learning that has usually been performed by the designer of a fixed-bias system. This new emphasis on a changing bias is an important research focus for digital libraries.

2.3 Associating text with visual features

Before proceeding, let's consider an example of retrieval in the context of video libraries of television shows. Consider the request, "Find comedians taking lie-detector tests." Like most requests, this one starts out with language, and will rely heavily on language-processing retrieval tools such as the caption-based retrieval tools of [5], [6]. However, the text had to get there somehow for these tools to be of use. Let's examine first some of the kinds of high-level text that tend to be available to help vision algorithms, and second, the use of vision-algorithms for helping generate text that can make the search easier.

2.4 Scripts and closed captioning

Figure 2 shows two frames from the "Seinfeld" TV series episode "The Beard." The frame at left shows the closed-captioning, where Cathy is saying "a polygraph." Figure 3 shows excerpts from the script corresponding to the two frames shown in Fig. 2.

The script can be used in several ways to simplify vision-based retrieval. First, the script tells which main actors will appear in a scene. Although they will not necessarily appear in every shot in that scene, this information can be used to bias the algorithm to try to verify first those actors listed in the script. Second, the script gives the location of the scene. Although this does not imply the backgrounds in each shot will be the same (cf. the two frames in Fig. 2) it is likely that certain objects in the scenes will reappear (cf. the green and white walls in the police station) and therefore become recognized as a feature associated with that set. Sometimes the script gives action information, although eye-catching actions are not always annotated in the script, but may be ad-libbed.

Closed-captions, intended for hearing-impaired viewers, can also be used to help vision algorithms associate actions with actors. The closed-captioning text is

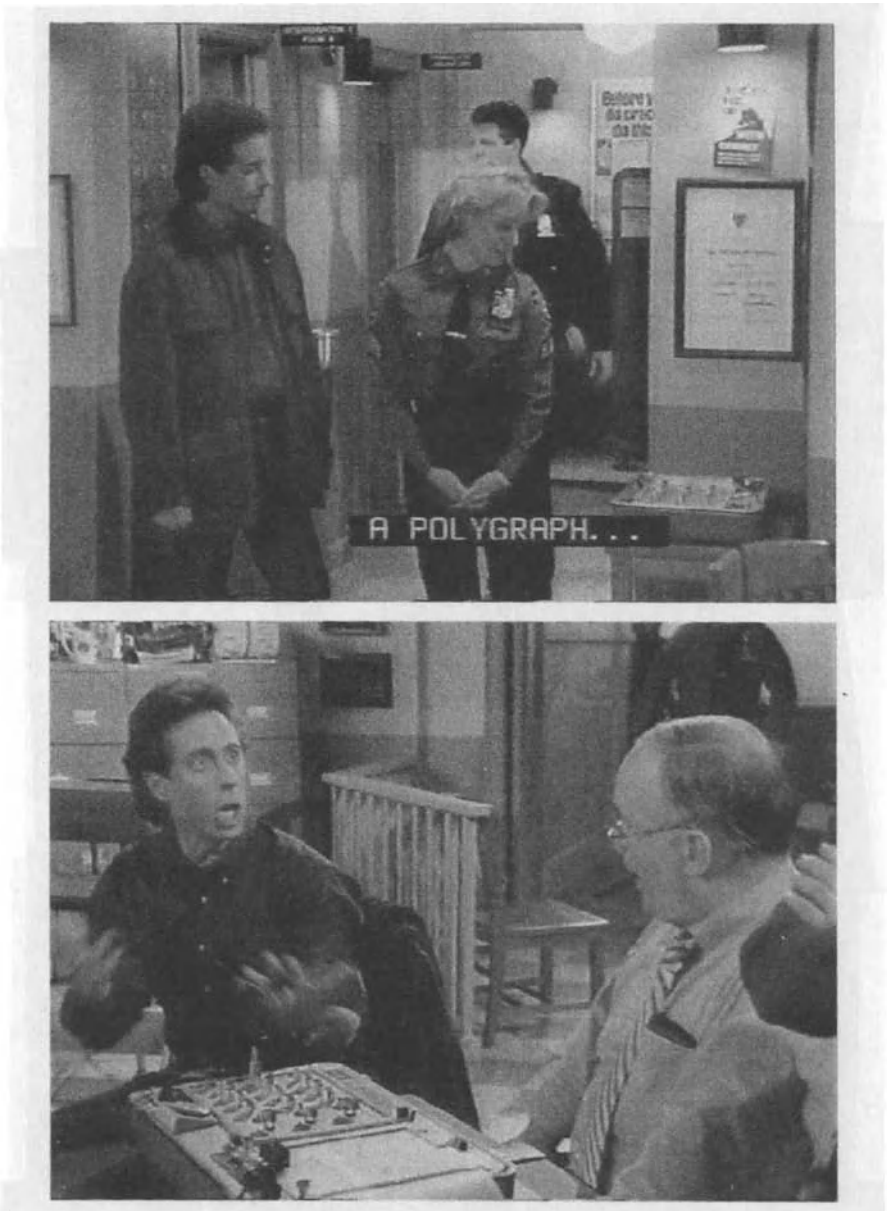


Fig. 2. Top: Shot where script mentions a polygraph. (See script excerpts in Fig. 3.) Note that the position of the closed-captioning text indicates which speaker (Cathy) is talking. Bottom: Shot where Seinfeld is hooked to a polygraph but the word "polygraph" and its synonyms such as "lie detector" do not appear anywhere in the scene containing this shot. Hence, script-based retrieval can get us near the answer, but vision is needed to find the answer. These images from "The Beard" episode of "Seinfeld" appear courtesy of Castle Rock Entertainment.

```
(Jerry, Cathy, Lou)
ACT ONE, SCENE N
INT. POLICE STATION HOUSE - DAY (2)
JERRY WALKING AROUND STATION HOUSE WITH CATHY.

JERRY
What's that?
CATHY
Polygraph.  What you civilians call a lie
detector test.
JERRY
Oh, Alright.  Let me ask you now when
someone is lying, is it true that their
pants are actually on fire?
CATHY
If I could tell you the famous faces that
have been up here.
...


(Jerry, Gus, Cathy)
ACT TWO, SCENE Y
INT. POLICE STATION - DAY (4)
JERRY IS WIRED UP TO A MACHINE THAT IS BEING
RUN BY GUS.  CATHY SITS NEXT TO HIM.

GUS
What's your name?
JERRY
Jerry Seinfeld.
GUS
What is your address?
JERRY
129 West 81st Street.
GUS
Did Kimberly steal Joe's baby?
...
```

Fig. 3. High-level information is available to work with low-level vision tools in most video productions today. The script says where and when the scene is set, and which main actors are present, but otherwise says little about scene content. These excerpts from "The Beard" episode of "Seinfeld" appear courtesy of Castle Rock Entertainment.

almost always a subset of the script, but its placement onto the image gives additional information. For example, the text appears closest to the person speaking if they are in the shot, and closest to their direction off the frame if they are not. The decoded closed captions can be located in the script to verify the name of the actor seen speaking. (Alternatively, a computer vision algorithm that recognizes the actors could assist a human in placing the captions for the hearing-impaired viewers.) The closed captions also display information that may not be in the script such as "knock knock" when somebody is at the door, or "ha ha" when it is not clear visually that one of the actors is laughing. These captions can also significantly aid in acoustic scene analysis.

2.5 Retrieval with low- and high-level learning

Now, let's return to the example request, "Find comedians taking lie-detector tests." Suppose that after going through a text thesaurus and other database tools, the retrieval system has determined that this episode of Seinfeld contains both the lie-detector synonym "polygraph" and the comedian Jerry Seinfeld. However, in the only scene where Jerry and the polygraph are both present according to the script (Fig. 2, left frame), he is not being given the test.

Several options are available at this point – one could watch the whole episode to see if he uses the polygraph, one could make a new query, or one could give up. With the aid of a smart-fast forward mechanism one could save time and watch just the scenes with Jerry in them. Alternatively, the user could combine some low-level vision into the query at this stage, by taking the frame at the left in Fig. 2, indicating to the learning system that the silvery box at lower right is of interest (query by example) and then asking the vision system to find all shots where that box appears with Jerry. The learning system determines which groupings best characterize the polygraph in the scene where the user labels it, and then uses these groupings, plus any prior knowledge about the data, to search for additional (unlabeled) appearances. Although general recognition of a polygraph is nearly impossible without, perhaps, a visual database of such devices (they come in many shapes, colors, and sizes), recognition of this particular polygraph can be accomplished, and used to retrieve frames such as the one at right in Fig. 2.

Moreover, after the retrieval has succeeded, the high-level polygraph label can be linked to the successful low-level visual features and stored, so that future queries for that situation will converge more quickly to the answer.

Although this is just one example, it illustrates several important points. First, there is no need to do "bottom-up" vision to label every segment in every image – this problem is notoriously ill-posed. (e.g. What *is* that stuff behind Cathy and Jerry? How should the railing be segmented or should it not be?) For decades, researchers have tried to generate edge-detection and segmentation outputs that are "right" even though there is no segmentation that people would agree on. Low-level vision should not proceed without high-level input.

FourEyes's gets around this problem by collecting groupings (leftmost stage) from which multiple segmentations can be constructed. Only after the user in-

teracts with the imagery, identifying regions of interest, do selections get made from the groupings, and receive labels.

Another point illustrated by this example is that most of what is in these scenes is unimportant compared to the actors. When a visual item is important, it is likely to either be named in the script at some point, centered in the camera, or gazed at by an actor; a high-level production mechanism is used to draw attention to it.

All of these high-level production mechanisms simplify the visual retrieval problem; their detection is another important challenge to vision researchers. However, their detection still does not solve the retrieval problem of finding queried content. Although closed captions associate speech with speakers, there is still no mechanism for associating labels with objects, actions, and overall scene mood. The associations linking high-level text with low-level image features still need to be learned.

The digital library is an important meeting place for high-level and low-level vision. Having the human in the loop, and working simultaneously with scripts, closed captions and low-level visual features, provides a unique environment for learning vision, especially for learning links between low-level features and high-level descriptions.

3 Summary

Digital video and image libraries are not only a new application area for computer vision, but they provide a unique opportunity to conduct research linking high-level and low-level vision.

Because retrieval tools for digital libraries involve a human user, vision algorithms in this domain have the opportunity to learn from human vision and interaction. I have mentioned one prototype system, "FourEyes," that attempts to do continuous learning, and have walked through an example of how it is applied to real problems, problems demanding the linking of text descriptions with visual features.

Retrieval systems that can also collect annotations and learn their links to the visual features, are not only of immediate use in applications, but they can potentially gather a great amount of useful data to researchers. In particular, the large number of mappings which result between relatively low-level visual features and high-level descriptions will be interesting to study for stability, context-dependence, and user-dependence.

The most important issue I hope to have drawn attention to is that of the need for more research on learning – and not directed to the one-time training of neural network weights or development of a set of discrimination functions, but to a system that learns continuously, switching its bias depending on the data and goals at hand, and updating that bias for long-term performance improvements.

Continuous learning for vision is a difficult area of research with far-reaching implications. Digital libraries provide a fertile arena for learning with a user in the loop, learning which high-level descriptions and low-level visual features are

best associated, and ultimately learning how to retrieve the most relevant and interesting data.

4 Acknowledgements

FourEyes is the thesis research of Tom Minka, and the script-based retrieval examples are drawn from the thesis research of Joshua Wachman. I would like to thank both Tom and Josh for numerous fun discussions which have influenced my ideas in this area. I am grateful to HP Labs, BT PLC, and NEC, for their sponsorship of this research.

References

1. D. B. Lenat, "Artificial intelligence," *Scientific American*, pp. 80–82, Sept. 1995.
2. R. W. Picard, "Light-years from Lena: Video and image libraries of the future," in *IEEE Second Int. Conf. on Image Proc.*, (Washington, DC), Oct. 1995. To appear; Also appears as MIT Media Lab Perceptual Computing TR #339.
3. A. R. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain.* New York, NY: Gosset/Putnam Press, 1994.
4. T. P. Minka and R. W. Picard, "Interactive learning using a 'society of models'," *Submitted for Publication*, 1995. Also appears as MIT Media Lab Perceptual Computing TR#349.
5. A. S. Chakravarthy, "Toward semantic retrieval of pictures and video," in *RIAO'94, Intelligent Multimedia Information Retrieval Systems and Management*, (New York), pp. 676–686, Oct. 1994.
6. R. K. Srihari, "Combining text and image information in content-based retrieval," in *IEEE Second Int. Conf. on Image Proc.*, (Washington, DC), Oct. 1995. To appear.

Computer Vision in Next Generation Image and Video Coding

Thomas S. Huang and Ricardo Lopez

University of Illinois at Urbana-Champaign - Beckman Institute
405 N. Mathews Ave., Urbana, IL 61801
huang@ifp.uiuc.edu, rlopez@ifp.uiuc.edu

Abstract. In this paper we will explore the various ways in which Computer Vision techniques can aid and sometimes even replace the classical algorithms used in many still image and video coding standards. Because compression factor is one of the most important performance criteria of image/video coding we will examine the effectiveness of these techniques in achieving low and very-low bit rates. Motivation for the recent demand of very low-bit rate algorithms stems in part from the introduction of advanced multimedia applications. The desktop computer, for instance, is fast becoming much more than a number crunching machine and will likely become the main channel for interpersonal communications as well as our interface to what is being called the *information superhighway.* The video inputs to such systems will not only need to capture and code images and video but also obtain some measure of understanding of the scenes before them. Classical or *first generation* coding algorithms, while indispensable in many fields of communications, fall short of this task. Only a fusion of such algorithms with the work currently being done in computer vision, image understanding, and artificial intelligence will produce a viable system. Since, traditionally, these communities have been rather disparate, it is necessary to begin at a basic level and explore the ways in which techniques from each field can be integrated to create a system that is truly more than the sum of its parts. With that end in mind, this paper will address some of the common aspects of image and video coding with a concentration on one of the more important research areas - the estimation of 2D and 3D motion to use in efficiently compressing video sequences using 3D models.

1 Introduction

This paper will be mostly an exploratory look and a discussion on the ways in which emerging computer vision techniques can aid or replace traditional still image and video coding algorithms. Our focus will be on the efficient and intelligent compression of video for the purpose of multimedia applications such as video teleconferences. The area of video compression has undergone significant

* This work was supported by National Science Foundation Grant IRI-89-08255, a grant from Texas Instruments, and a Fellowship from AT&T Bell Laboratories.

advancement in the past few decades, however, it is thought by many to have reached a plateau in its reduction of bit rates for standard image sequences. Current video coding standards are not able to provide good quality results at the very low bit rates envisioned by many for use in future interpersonal communication systems [1]. This has prompted the proposal of an MPEG-4 standard, bringing forth several clever algorithms that fuse together ideas from image processing, computer vision, and image understanding. Undoubtedly, the very low bit-rate video coders of the near future will require this type of cohesion to provide robust and efficient communication interfaces. In the following sections we will examine the problems in still image and video sequence coding. While we will touch briefly on many subjects, a concentrated effort will be made in examining the estimation of 3D motion from scenes for use in model-based video coding.

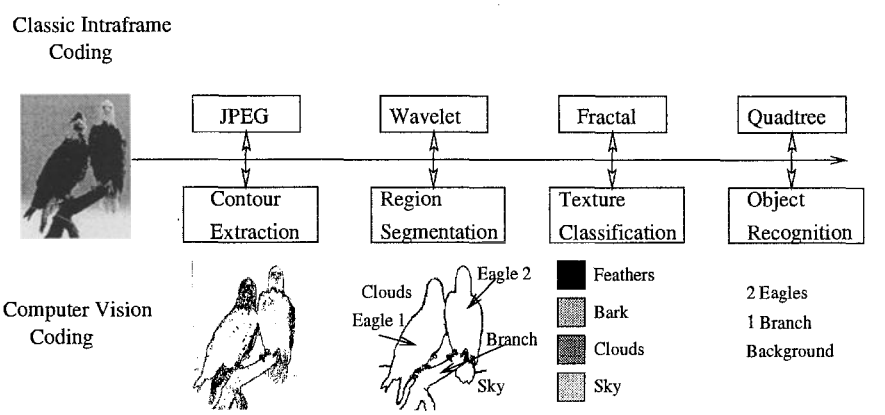


Fig. 1. Classic still image coding versus Intelligent Computer Vision coding

2 Still Image Coding

Traditional still image coding requires the removal of spatial redundancies by reducing the correlation within an image [2]. The most common algorithm is described by the JPEG standard and is in wide use today [3]. There has been a wealth of research devoted to improving on the JPEG algorithm with respect to both quality and compression ratios. More advanced algorithms involving wavelets, subband decompositions, quadtree based structures, etc. have been devised. However, many of these improvements have been limited by the underlying methodology used in classical image coding which deals with images at the pixel or block level [4], [5].

With the emergence of Computer Vision out of its infancy during the past decade, there have been an increasing number of reliable algorithms that re-

searchers have begun applying to the problem of image coding. Basic techniques such as segmentation, contour extraction, texture modeling and 2D object recognition can be combined in clever ways to drastically reduce the bit rate for still images, Fig. 1. The motivation is to base the coder on a higher level of scene representation, as opposed to conventional statistical image models. In Fig. 1, for example, standard image coders would divide the pixels into blocks, compute DCT transforms, quantize the coefficients, and entropy code them to obtain the final bit stream. Nowhere in this process is it taken under consideration that the scene contains 2 eagles, a branch, clouds, and sky. Successfully extracting this higher level of knowledge and representing it in such a way that a decoder would be able to reconstruct the image is a prevalent goal in using computer vision for coding. Possible representations include textures, contours, and shape descriptions. Object recognition algorithms, together with appropriate object-model databases at both encoder and decoder could also be used. The arguments against these types of approaches have been the unreliability of computer vision techniques to work with general classes of images. This stance, however, has been steadily decreasing in strength with the emergence of more stable and robust computer vision algorithms.

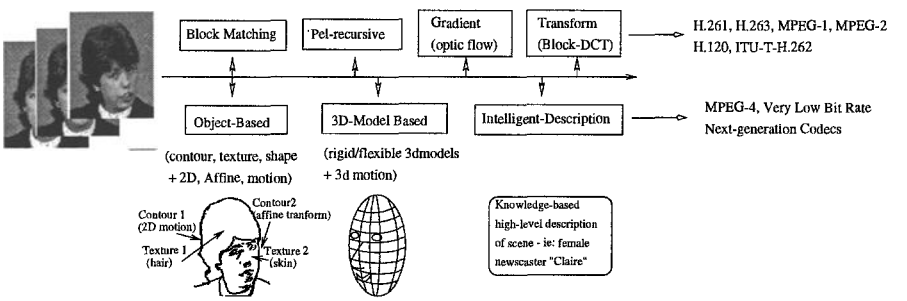


Fig. 2. Classic video image coding versus Intelligent Computer Vision coding

3 Video Coding

One of the most important applications of compression research has been in the field of video-sequence coding. The gains obtained by removing temporal as well as spatial correlation have been substantial and several standards have been introduced to deal with different video requirements [6], [7]. The recent demand for very low-bit rate algorithms, however, have found "classic" techniques lacking and researchers have begun looking to computer vision, graphics, and other fields for useful tools, Fig. 2. The target is good quality video at rates well under 64 kbs for applications such as video teleconferencing and personal communication devices, [8]. Next generation video codecs have been produced which are

either completely based on "non-traditional" video coding techniques or combine existing standards such as H.261 and H.263 with computer vision/graphics techniques [9].

One of the major contenders for use in very-low bit rate systems is model-based coding. This field is steadily growing and some excellent reviews can be found in the literature [10], [11]. Most of the research has been centered on the modeling of human heads and faces because of the obvious advantage in personal communication systems. Among the challenges in implementing such a coding system are human head and face modeling, expression analysis/synthesis, feature extraction and head tracking. This last area is of particular interest and involves not only estimating the 3D global motion of the speaker, but also the local, non-rigid motion due to expressions, speech, and skin deformations [12], [13].

4 Feasibility of a Model-Based Video Coding System

While model-based coding for video applications can be considered on the cutting edge of video compression technology, the field is still in its infancy. There has been debate on what type of benefits such systems may be able to provide in the short term, especially with the development of MPEG-4 in full swing. Since the individual components of model-based systems involve many unsolved problems, there has been a notable lack of experiments in building and testing end-to-end real-time systems. The research to date has been extensive, but modular in its approach. With this in mind, a significant research effort at Beckman's image lab has begun to explore the performance and feasibility of a real-time, end-to-end system. This section explores the major obstacles of such an undertaking and reports the progress made.

4.1 System Overview

Implementing the system shown in Fig. 3 will require the synergism of many different areas of computer vision and image processing. This, however, will provide the opportunity to test a significant portion of our research in a "real-world" setting. Recent advances in head-modeling, pose estimation, feature identification and tracking, and emotion analysis will all be implemented. We can breakdown the system into the following major modules, with a brief description of possible approaches:

- 3D Head modeling
- Feature location/tracking
- Global head pose estimation
- Eye, lips motion
- Synthesis of video from 3D model

Eventually, the goal will be to operate all these modules automatically, but in the short-term user interaction may be necessary. Current options for the head-modeling stage include the fitting of generic wireframes to several 2D images

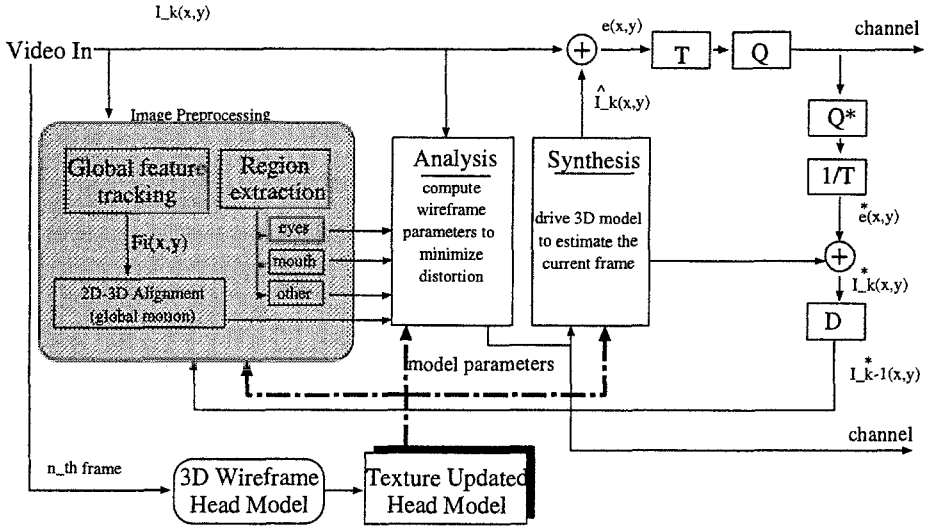


Fig. 3. Layout of a proposed real-time model-based coding system.

of a subject. Another possibility is the use of 3D range data from a Cyberware Head Scanner. For feature location and tracking standard template matching and optical flow techniques are being considered. Head pose estimation is currently being perform using 2D-3D feature point matchings to produce an alignment transform that approximates the head orientation. Some preliminary results are available and will be presented in the next section. One of the more difficult aspects of the system is determining the local deformations due to eyes and lips and representing these in a compact way for the subsequent synthesis. Two main options include either tracking small feature sets which encompass the general shape of the eyes and lips, or extracting the contour of these regions using techniques such as snakes and edge detection.

Synthesis of the video sequence at the receiving end will be accomplished by driving the 3D head model with the small set of parameters obtained in the previous stages. Periodic texture map updates will be used to account for significant lighting and scene changes. The system shown in Fig. 3 resembles the typical motion-compensated hybrid coding system, except for the generation of the current frame estimate [14], [15]. Instead of calculating optimal 2D motion vectors and using them in the prediction, the input images, aided by the 3D head model, are used to calculate appropriate model parameters for the analysis-synthesis stage. These parameters are sent down the channel along with the current frame's residual for use in the reconstruction at the decoder. One open research area being considered is the formulation of a new difference criteria appropriate in computing this residual. Also, the intelligent use of the 3D model

in motion prediction is being investigated. Research in these areas will help us develop a parallel to conventional motion-compensated video coding using a model-based framework.

Fig. 4. Testbed for experimenting with model-based video coding.

4.2 Preliminary Results

This section will describe in more detail the head-pose estimation module in the system proposed above. Compensation for global motion is necessary in accurately detecting the local deformations of the human face. It is also one of the major benefits of model-based coding to encode in a very compact form the large variations in image intensities that occur during large global movements. Conventional image compression techniques requires much larger amounts of information to code this type of movement. The testbed currently being used is shown in Fig. 4 and simulates a real-time teleconferencing environment. The left window is used to display the 2D video sequence and the results of feature point tracking, eyes, lips detection, etc. The right window is used to display the results of driving the 3D head model with the recovered parameters and texture map then projecting it into the image plane.

The technique currently being implemented involves the computation of an alignment transform from pairs of 2D-3D feature points [16], [17]. The approach only requires 3 point matches, so tracking a larger feature set enables us to choose the triple of points that give the best pose estimate. This increases the robustness of the system with only a slight performance cost. There are many techniques available that provide pose estimation of 3D objects from 2D images, but the requirements of a real-time system restrict us to methods which are computationally efficient.

Basically, the algorithm we implement assumes a group of features, (F_{2D}, F_{3D}), are available in the 2D image and in the 3D model and that correspondence is known. We then take subsets, f, of this feature group which contain 3 point pairs at a time. Using the algorithm described in [16], a series of transforms are computed involving a rigid transform, R, and an affine transform, A, that aligns the 3D features in f to their 2D counterparts. The resulting transform, $T(R, A)$, is then applied to the entire feature set F, and the error in the subsequent alignment is determined. This error is calculated by measuring the error between

a transformed 3D point in F to its corresponding 2D feature. After rejecting abnormally large errors which would be caused by loss of a feature during 2D tracking, the RMS error for the entire feature set is calculated for each transform $T(R, A)$. Applying this procedure for all unique f's in F and choosing the transform which results in the smallest error provides the best alignment, or pose, estimate for the 3D object. In other words, our goal is to solve:

$$\min_{f \in F} Error(T(R, A) \circ F_{3D}, F_{2D})$$

and use the resulting f_{min} to compute the best pose estimate.

This part of the system has been tested with real 2D video sequences, using a texture mapped 3D range model for image synthesis. Although the algorithm has an underlying orthographic + scale imaging model assumption, it appears to deal well with the error introduced by perspective effects. The original frames and their synthesize equivalents are shown in Fig. 5. Future and ongoing work will include adding facial deformations such as smiling and blinking to the 2D video input.

5 Conclusion

In this paper we have explored some of the different ways in which computer vision techniques can be used to produce more intelligent image/video coding systems. Areas such as segmentation, contour extraction, texture analysis, and 3D modeling can provide useful tools to build the video coding systems of the next generation. By combining such techniques with classic coding standards it will be possible to break the 64 kbs barrier and still provide robust, good quality images and video sequences. We have proposed a real time end-to-end video conferencing system and described some of the modules required for such an effort. Preliminary results for the global head-pose computation portion of the system were presented and were shown to provide accurate estimates of head orientation at low computational cost.

References

1. David W. Lin, Cheng-Tie Chen, and T. Russell Hsing. Video on phone lines: Technology and applications. In Zhang et al. [18], pages 175–192.
2. A. Netravali and B. Haskell. *Digital Pictures, Representation, and Compression.* Plenum Press, New York, 1988.
3. G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34:30–44, April 1991.
4. Thomas W. Ryan, L. Darwin Sanders, and H. Donald Fisher. Wavelet domain texture modeling for image compression. In *First IEEE International Conference in Image Processing* [19], pages 380–383.
5. Dinei A. F. Florencio and Ronald W. Schafer. A non-expansive pyramidal morphological image coder. In *First IEEE International Conference in Image Processing* [19], pages 331–335.

Fig. 5. Head pose computation results. The left column in each figure is the original video sequence. The right column is the synthesized image using the 3D model and the computed pose. (a) Frames 1-7 (b) Frames 20-25 (c) Frames 50 - 55.

6. Ralf Schafer and Thomas Sikora. Digital video coding standards and their role in video communications. In Kunt [20], pages 907–924.
7. Rangarajan Aravind, Glenn L. Cash, et al. Image and video coding standards. *AT&T Technical Journal*, pages 67–89, January/February 1993.
8. Touradj Ebrahimi, Emmanuel Reusens, and Wei Li. New trends in very low bitrate video coding. In Kunt [20], pages 877–891.
9. M. F. Chowdhury, A. F. Clark, A. C. Downton, E. Morimatsu, and D. E. Pearson. A switched model-based coder for video signals. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):216–227, june 1994.
10. Donald E. Pearson. Developments in model-based video coding. In Kunt [20], pages 892–906.

11. Kiyoharu Aizawa and Thomas S. Huang. Model-based image coding: Advanced video coding techniques for very low bit-rate applications. In Zhang et al. [18], pages 259–271.
12. T. Fukuhara and T. Murakami. 3-d motion estimation of human head for model-based image coding. *IEE Proceedings*, 140(1):26–35, February 1993.
13. Haibo Li, Pertti Roivainen, and Robert Forchheimer. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.
14. Chi-Fa Chen and K. K. Pang. Hybrid coders with motion compensation. In *Multidimensional Systems and Signal Processing*, volume 3, pages 241–266. Kluwer Academic Publishers, Netherlands, 1992.
15. Bernd Girod. The efficiency of motion-compensating prediction for hybrid coding of video sequences. *IEEE Journal on Selected Areas in Communications*, SAC-5(7):1140–1154, August 1987.
16. Shimon Ullman and D. P. Huttenlocher. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.
17. Ricardo Lopez and Thomas Huang. Head pose computation for very low bit-rate video coding. In Vaclav Hlavac and Radim Sara, editors, *Computer Analysis of Images and Patterns*, pages 440–447, Prague, Czech Republic, September 1995. Springer.
18. Y.-Q. Zhang, W. Li, and M. L. Liou, editors. *Proceedings of the IEEE: Special Issue on Advances in Image and Video Compression*, volume 83, Feb 1995.
19. IEEE. *First IEEE International Conference in Image Processing*, Texas, USA, November 1994. IEEE Press.
20. M. Kunt, editor. *Proceedings of the IEEE: Special Issue on Digital Television*, volume 83, June 1995.

Video Annotation: Computers Watching Video

Aaron F. Bobick

MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 USA
bobick@media.mit.edu
http://www.media.mit.edu/~bobick/

Abstract. Recently our research has focused on tools for video annotation — the generation of symbolic descriptions of dynamic scenes. In this paper I describe two specific examples of developing capabilities necessary for the understanding of action in video sequences. The first is a novel tracking technique that builds context-specific templates that are used only locally in time and space. The second example is some work on gesture recognition where time is implicitly represented in a probabilistic manner. We believe the future of computer vision lies in the processing of video and that while much work has been devoted to the representation and manipulation of static images, we are far behind in developing tools for considering action.

1 Video Annotation

Video annotation is different than conventional image understanding in that one is interested in what is *happening* in a scene, as opposed to what *is in* the scene. In some regard, one has to be crazy to pursue research in video annotation. After all, if decades of image understanding research have made only small inroads in being able to describe a single picture, one might believe that attempting to describe what is happening in hundreds of images is not a viable research of goal.

However, video annotation can be regarded as a way of providing *more* constraint in the interpretation of imagery. Working in constrained domains, we require that the image interpretation be plausible over time; extracted structure must obey the temporal constraints of the domain. For example, if we are annotating an American football play, we might be interested in tracking the quarterback. Unfortunately, current (even near future) technology cannot see or track the quarterback in every frame. However, assuming he never disappears from the field of play, we can "track" him as he enters an amorphous blob and re-emerges six frames later. The program cannot see him during this time, but it knows he's there.

Understanding time can be either explicit, as in the above example, or implicit, captured in the representation of action. One example that we will expand upon later is our work in gesture recognition [[1, 2]]. In this work gesture is represented either deterministically by an explicit sequence of states through which

the hand must move, or probabilistically by a hidden Markov model. In both cases the requirement that the interpretation be consistent with the temporal constraints of the domain is guaranteed by matching the input data to learned representations of action which are sensitive to time.

From our perspective, one of the future directions of computer vision lies in the area of video annotation, generating the descriptions of action. In this paper I will present two different aspects of processing "action" sequences. The first focuses on exploiting the continuity of time by constructing "context-specific" templates that are valid only locally in time and space. The second effort describes some recent work in gesture recognition which develops an implicit model of time captured in a probabilistic framework. Finally, in the conclusion I will refer to additional work where knowledge about time and actions is explicitly expressed in rules that are used by the system to interpret the imagery.

2 Locality in time and space: tracking

Many video annotation domains require documenting the interactions between people and other non-rigid objects against non-static backgrounds and in unconstrained motion. Examples include sporting events, air traffic, pedestrian mall traffic, or cell movements from quantitative fluorescence microscopy. In this section we describe a technique that incorporates contextual information into a low-level tracking to successfully recover the trajectories of such objects.

The domain in which we explore the problem is American football. Figure 1 illustrates the large amount of pan and zoom present in a typical game film of a football play, where the cameraman must keep as many players as possible in the field of view.[1] The camera pan- and zoom-rates, the wide focal-length, the de-interlacing, the erratic and colliding behavior of the players — all of these complexities yield a situation in which traditional tracking algorithms fail; for a more thorough consideration see [3].

2.1 Closed-worlds

The task of tracking objects in a complicated domain such as football requires using some type of knowledge about the world, or *context*. Here we consider the context of a tracking problem to be a boundary in the space of knowledge — a boundary outside of which knowledge is not helpful in solving the tracking problem. Continuing in football, a context would be something like "a region of the field near the upper hashmark on the 50 yard line that contains two players, one offensive and one defensive."

To use context effectively in tracking, we exploit a *closed-world* framework. A closed-world is a region of space and time in which the specific context is adequate to determine all possible objects present in that region. Several closed-world image regions are shown in Figure 2. The regions contain turf, a line,

[1] For more data, results, and general cool football stuff see:
http://www-white.media.mit.edu/vismod/demos/football/football.html.

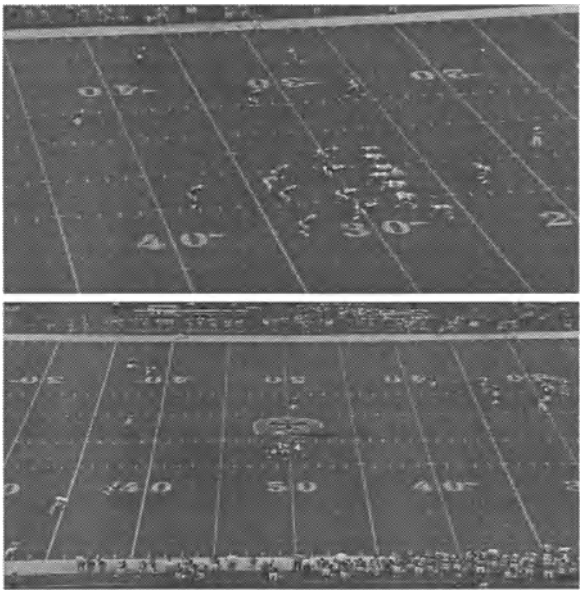


Fig. 1. Two 700x240 deinterlaced frames of a typical pass play from pan/zoom video of a college football game.

hashmarks, one or two players, and/or part of a logo object. The internal state of the closed-world — e.g. the positions of the players — is unknown and must be computed from the incoming visual data. Visual routines for computing the internal state can be selected using the context-restricted domain knowledge and any information that has already been learned about the state within the world from previous processing. For example, intensities can be used to separate offensive from defensive players; texture can be used to separate players from backgrounds. We note that a few other authors have developed systems that use contextual information and/or approaches similar to our closed-worlds [4, 5].

2.2 Context-specific templates

Given a closed-world we can construct a template of the player to track the player to the next frame. A closed-world is shown in Figure 3-a, and bitmap representations of the models of the objects in that closed-world are shown in Figure 3-b. The actual models used for all these objects are slightly different (i.e. the histogram ranges and variances are all different). Since this closed-world contains only one player a simple pixel removal scheme is used to construct the template. Each background object can explain any plausible pixels, marking them as "don't care." The remaining "player pixels" are used to construct the template shown in Figure 3-c. This template is the context-specific feature used

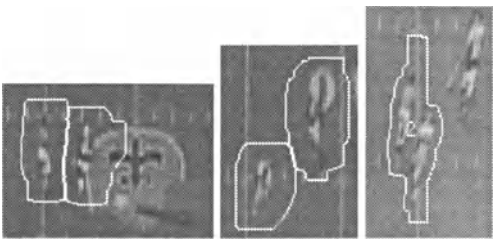


Fig. 2. Three different closed-world regions. The rightmost closed-world contains two players and line and hashmark objects.

to track the player in the next frame. The correlation score is shown in Figure 3-d. For each object, a new template is constructed in each frame, using the features and pixels determined by the closed-world analysis.

One important aspect of the closed-world tracking is that the features being tracked (the context-specific templates) are valid only locally in time and space. The templates created are local in space in the sense that they can only be applied in a region near the last position of the player; if applied everywhere in the image they would probably have many false alarms. They are local in time in that after a few frames they are no longer valid for tracking.

We have extended the above technique to handle the case of two-players in a closed-world region. The basic approach remains the same: use knowledge of characteristics of the objects to assign the pixels. However, the now the system uses the knowledge about the two players to select image attributes to disambiguate between the players.

The algorithm described has been tested on several football plays; typical duration is 270 frames. An example of tracking a single player through 200 frames is shown in figure Figure 4-a and Figure 4-b. Because the template does not contain many contaminating pixels from the background objects, the field markings do not come to control the behavior of the template. In Figure 5 we have overlayed some of the recovered player paths on a background image of the field obtained using median filtering over the entire image sequence. These paths are the type of input that will be used by a play understanding video annotation system we are developing. The goal is to label the plays based upon the actions of the players.

3 Implicit time: gesture recognition

The possibility of machine understanding of human movement and gesture has brought new possibilities to computer-human interaction. Such interest has inspired research into the recovery of the the complete 3-dimensional pose of the body or hand using a 3-dimensional physical model (e.g. [6, 7]). The presump-

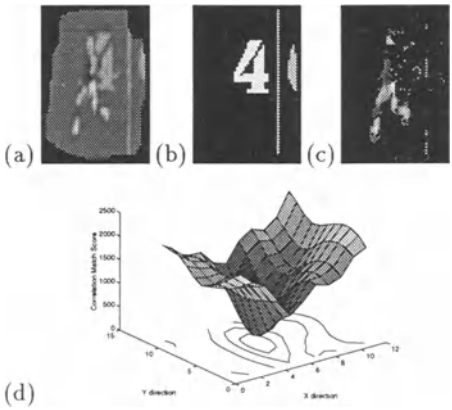


Fig. 3. (a) Closed-world image region. (b) Objects in closed-world. (c) Remaining template after pixels are removed by being "possibly-explained" by the objects. (d) A typical correlation match score for the next frame.

tion behind such work is that a complete kinematic model of the body will be required for useful inferences.

We claim, however, that constraints exists that may make full kinematic recovery of 3-dimensional pose unnecessary. First, we note that gestures are embedded within communication. As such, the gesturer typically orients the movements towards the recipient of the gesture. Visual gestures are therefore *viewpoint-dependent.* [8, 9] Second, in the space of motions allowed by the body's degrees of freedom, there is a small subspace that we use in the making of a gesture. Taken together, these observations argue for a view-based approach in which only a small subspace of human motions is represented.

How should a system model human motion to capture the constraints present in the gestures? There may be no single set of features that makes explicit the relationships that hold for a given gesture. In the case of hand gestures, for example, the spatial configuration of the hand may be important (as in a point gesture, when the observer must notice a particular pose of the hand), or alternatively, the gross motion of the hand may be important (as in a friendly wave across the quad). Quek [10] has observed that it is rare for both the pose and the position of the hand to simultaneously change in a meaningful way during a gesture.

Recently we have presented an approach that represents gesture as a sequence of states in a particular observation space [1]. We then extended that work and developed a technique for learning visual behaviors that 1) incorporates the notion of multiple models — multiple ways of describing a set of sensor data[11]; 2) makes explicit the idea that a given phase of a gesture is constrained to be within some small subspace of possible human motions; and 3) represents time as a probabilistic trajectory through states [2]. The basic idea is that the different

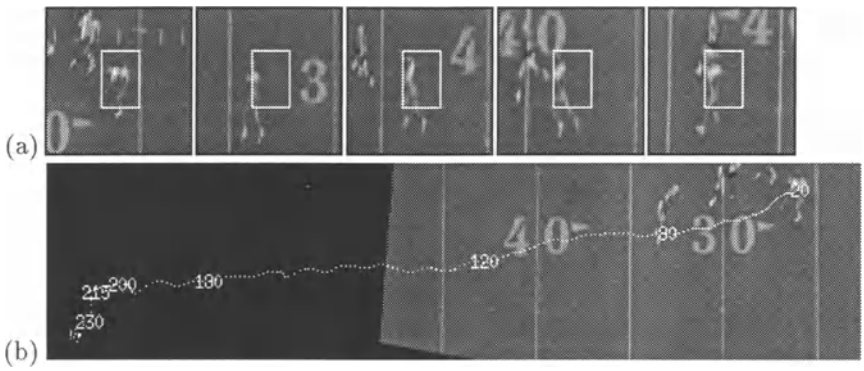


Fig. 4. (a,b) Successful closed-world tracking through frame 200, despite path over field objects.

models need to approximate the (small) subspace associated with a particular state and membership in a state is determined by how well the state models can represent the current observation. The parsing of entire gesture is accomplished by finding a likely sequence of states given the memberships and the learned transition probabilities between the states.

The details of the techniques are presented in [1, 2]. The approach is based upon state models that define a *residual* — how well a given model can represent the current sensor input. We then embed this residual-based technique within a Hidden Markov Model framework; the HMM's represent the temporal aspect of the gestures in a probabilistic manner and provide an implicit form of dynamic time warping for the recognition of gesture.

Here we illustrate the technique by way of two examples. Figure 6 — a wave gesture — consists of a single model example but shows the use of the HMM. In this case, the parameters associated with a model instance are simply a number of the top eigenimages that account for most of the variance of the training images (as indicated by the eigenvalues) and the mean training image. The input consists of 32 image sequences of a waving hand, each about 25 frames (60 by 80 pixels, gray-scale) in length.

The recovered Markov model, the mean image at each state, and plots of the memberships and residual for one training sequence are shown in Figure 6. The recovered Markov model allows the symmetry of motion seen in the plot of membership over an observation sequence. Some other observation sequences differ in the extent of the wave motion; in these cases the state representing the hand at its lowest or highest position in the frame is not used.

Our second example describes the position and configuration of a waving, pointing hand (Figure 7). In each frame of the training sequences, a 50 by 50 pixel image of the hand was tracked and clipped from a larger image with a clut-

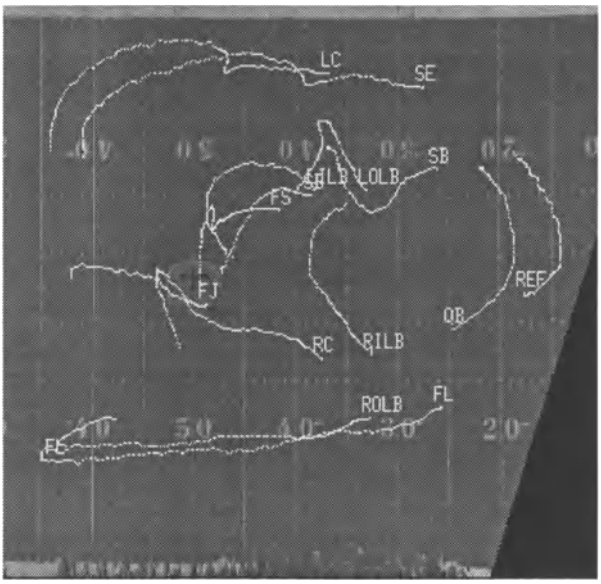


Fig. 5. All recovered paths overlayed on an image of the field. These paths will be used as input for a video annotation play understanding system.

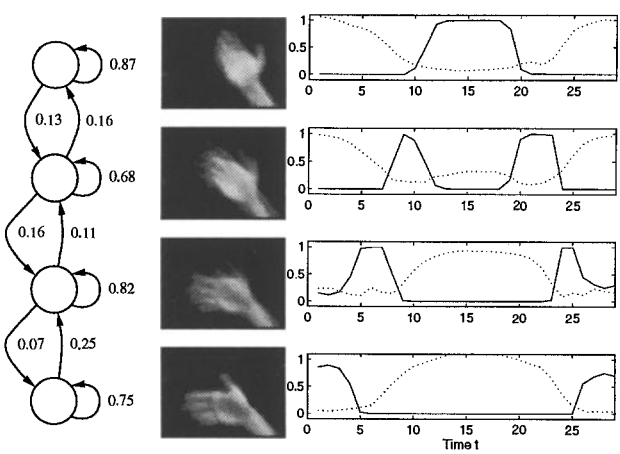


Fig. 6. A wave gesture. The recovered Markov model for all training sequences at left shows the symmetry of the gesture. The mean image for each state is shown in the middle. On the right is a plot of membership (solid line) and residual (dotted line) for each state for one training sequence. The exact shape of the plots varies in response to the variance and length of the sequence.

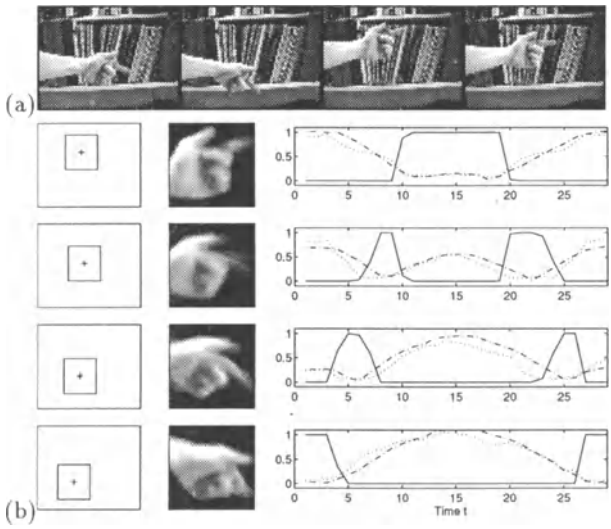


Fig. 7. (a) Four representative frames (ordered left to right) are shown from one training sequence. (b) The mean location of the tracked hand in the larger image is shown on the left. The mean image for each state is shown in the middle. On the right is a plot of membership (solid line), configuration residual (dotted line), and the position residual (dash-dotted line) for each state for one training sequence.

tered background. Foreground segmentation was accomplished using the known background. The configuration C of the hand is modeled by the eigenvector decomposition of the 50 by 50 images. The position P of the hand is modeled by the location of the tracked hand within the larger image. The recovered Markov model is similar to that of the waving hand in the previous example.

The variance of each feature indicates the importance of the feature in describing the gesture. In this example both the position and configuration of the hand was relevant in describing the gesture. Had the location of the hand varied greatly in the training set, the high variance of the position representation would have indicated that position was not important in describing the gesture.

4 Action in context

Finally, I mention some current work that makes time explicit. In [12] we introduce *SmartCams,* cameraman-less cameras that respond to a director's requests. Such cameras perform inverse video-annotation: given some symbolic description (*"close-up chef"*) the system needs to generate the correct image. One key element of the system is that it maintains an approximate world model to control the selection of view-based vision routines, and that selection process is controlled by rules that explicitly model time and action. For example, a bowl that was manipulated by the chef is known to be currently located where the chef

last placed it. For more details and a demonstration see:
`http://www-white.media.mit.edu/`
`vismod/demos/smartcams/smartcams.html`

5 The future

Computer vision has just begun to consider action. Yet, the vast majority of images recorded are frames of video sequences. For computer vision to begin to understand the images in our environment, or to understand the environment itself, it is clear that understanding action and behavior is fundamental. Computer vision has developed numerous ways of representing a cup (Euclidean solids, superquadrics, spline surfaces, particles); how many ways do we have to represent throwing a baseball?

Acknowledgment: Thanks to the entire HLV lab and in particular Stephen Intille, Andy Wilson, and Claudio Pinhanez who did most of the labor reported here. The work presented here is supported in part by a research grant from Interval Corp and by ORD contract 94-F133400-000.

References

1. A. F. Bobick and A. D. Wilson, "A state-based technique for the summarization and recognition of gesture," *Proc. Int. Conf. Comp. Vis.*, 1995.
2. A. D. Wilson and A. F. Bobick, "Learning visual behavior for gesture analysis," in *Proc. IEEE Int'l. Symp. on Comp. Vis.*, 1995. submitted for publication.
3. S. Intille and A. Bobick, "Closed-world tracking," in *Proc. Int. Conf. Comp. Vis.*, June 1995.
4. H.-H. Nagel, "From image sequences towards conceptual descriptions," *Image and Vision Comp.*, vol. 6, no. 2, pp. 59–74, 1988.
5. T. Strat and M. Fischler, "Context-based vision: recognizing objects using information from both 2D and 3D imagery," *IEEE Trans. Patt. Analy. and Mach. Intell.*, vol. 13, no. 10, pp. 1050–1065, 1991.
6. J. M. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: an application to human hand tracking," *Proc. European Conf. Comp. Vis.*, vol. 2, pp. 35–46, 1994.
7. K. Rohr, "Towards model-based recognition of human movements in image sequences," *Comp. Vis., Graph., and Img. Proc.*, vol. 59, no. 1, pp. 94–115, 1994.
8. T. Darrell and A. Pentland, "Space-time gestures," *Proc. Comp. Vis. and Pattern Rec.*, pp. 335–340, 1993.
9. Y. Cui and J. Weng, "Learning-based hand sign recognition," in *Proc. of the Intl. Workshop on Automatic Face- and Gesture-Recognition*, (Zurich), 1995.
10. F. Quek, "Hand gesture interface for human-machine interaction," in *Proc. of Virtual Reality Systems*, vol. Fall, 1993.
11. R. W. Picard and T. P. Minka, "Vision texture for annotation," *Journal of Multimedia Systems*, vol. 3, pp. 3–14, 1995.
12. A. Bobick and C. Pinhanez, "Using approximate models as a source of contextual information for vision processing," in *IEEE Workshop on Context Based Vision*, June 1995.

Video Representations

Ruud M. Bolle[1], Yiannis Aloimonos[2] and Cornelia Fermüller[2]

[1] Exploratory Computer Vision Group, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
[2] Computer Vision Laboratory, University of Maryland, College Park, MD 20742-3275, USA

Abstract. The recently emerging field of video analysis for the purpose of indexing, browsing, compression and recall through content-based queries is in need of a formal framework that can serve as the basis for algorithms. This paper argues that such a framework can be obtained by borrowing concepts from qualitative techniques in computer vision.

1 Introduction

More and more video is generated every day. Much of this data has been stored in some analog form such as VHS video or film. But now the trend is towards total digitization of video, and with the advent of cheaper digital storage devices, it becomes economically feasible to digitize the video data and store or transmit it in some sort of digital form. All storage and transport mechanisms to television receivers and computer displays will eventually be dominated by digital technology [1]. This includes CD-ROM, VTR, telecommunication networks, cable, terrestrial and satellite transmission, etc.

The digital form allows video to be processed using digital computers. This opens up possibilities for computer analysis of the video content to generate a number of appropriate representations that will contribute to a flexible video database organization enabling fast and reliable operations for indexing, browsing, editing, and compression. Video indexing and browsing refer to the operation of retrieving video shots from longer sequences based upon some sort of organization of the linear time-oriented structure of the video, or, perhaps more interesting, based upon the video content. For the latter case, examples of video content are the presence of objects in the video, the appearance or disappearance of objects, and actions or interactions of objects. Video editing includes the act of modifying the content of a video clip, e.g., removal or replacement of objects in a video clip. Many more video editing tools can be envisioned—for example, computationally changing the lighting conditions under which a video clip was filmed. Finally, video compression is the operation of lossy or lossless data reduction of video streams defined in standards such as MPEG. At this point, these operations may appear somewhat disjoint and in the literature they have been treated as distinct specific applications; however, the needed computational analysis for such applications is very much related to the analysis of video sequences performed by a moving visual observer. Using the visual data,

an observer has to develop a number of space-time representations that allow the accomplishment of a number of tasks. In the same spirit, we propose here that content-based video retrieval and manipulation can be achieved through a hierarchy of spatiotemporal representations developed from the video.

2 Previous Work

Currently, most video database organization is done by hand annotation through meticulous pre-viewing of the video or (possibly) by analysis of the closed caption. However, to scale to larger databases of video, automatic extraction of visual content information is desired. The first work on this problem was on generating temporal indices (organizing video into segments, scenes, shots). Tonomura [2] proposed scene cut detection based on frame histograms. Ueda et al. [3] use the rate of change of the histogram and consider the detection of zooming and panning of the camera by dividing the frames into non-overlapping small regions. Otsuji [4] presents a cut detection algorithm based directly on subsequent frame differences.

Given that the scene cuts are properly detected, an observation made by various researchers [5, 6] is that the representation of a video shot as a single still image would be a significant step toward the video indexing problem. Assuming there is no independent motion (only camera motion) in a shot, then by finding the transformation between subsequent frames, the frames can be "pasted" together into a larger image. If there is independent motion, the moving object(s) have to be segmented from the background and an image of the background can be reconstructed—along with images and trajectories of the moving objects. Techniques such as salient stills [5, 7, 8], mosaicking [9], and Video Space Icons [6] are proposed to obtain such still images. The rationale is that traditional image database indexing techniques such as [10] can be applied to the composted images and the segmented object images; motion-based indexes are also envisioned.

In this paper, we argue that even though scene cut detection, based on global or other frame properties, and mosaicking approaches, based on video formation models of various degrees of complexity, are useful and needed, often more fundamental analysis of the video is necessary. This analysis should not be restricted to 2D approaches, as it has been the case in the literature, but it should take into account that video is the result of the projection of a 3D world onto the 2D film.

3 A Hierarchy of Representations

If we aim to characterize video sequences, we have to come up with some methodology for studying this problem. The ideal thing would be to design a clearly defined architecture of a video analysis system and to start working on its components; however, we have few answers available when it comes down to actually talking about the representations that are relevant in the video handling

domain. Answers to these questions cannot come as insights gained from the study of mathematical models. It must be empirical studies investigating video sequences that will tell us how to couple video handling functions, video representation and video generation. Up till now we haven't understood how we actually could develop representations of videos depicting complex environments, so we won't be able to obtain a global view of the overall architecture and funtionality of a video analysis system. At this time it also wouldn't contribute much to the development of our understanding to just go ahead and develop a particular system that has a particular capability—say, for example, to build a retrieval system that finds video clips of tables. Even if we were able to create such a system that has a success rate of 99%, this system would have the capacity of recognizing many things that are unknown to us, and not just tables. Thus by aiming to build systems that compute certain representations that seem relevant to our symbolic language repertoire, we wouldn't gain much insight into video analysis.

It thus seems somehow natural that the only way out of this problem of where to start is to approach the study of video analysis in an "evolutionary" way. We call such an approach the synthetic (evolutionary) approach. The ideas behind this approach we will discuss in detail in the remainder of this paper. It means that we should start by developing individual primitive representations and provide the video handling system in this way with capabilities. As we go on, the representations will become more and more complex. At the same time, as soon as we have developed a small number of representations, we should work on their integration. Such an endeavor throws us immediately into the study of other major components of a video handling system: We are confronted with the study of spatial memory with all its associated problems of memory organization, visual data representation, and indexing—the problem of associating data stored in memory with new visual information.

The simplest capability in video handling is the one responsible for recognizing the 3D motion with which the camera was moving while capturing the video (also known as egomotion estimation capability). Because this is based on the rigid motion model, this is the simplest capability. Next in the hierarchy come the capabilities for localizing/segmenting independently moving objects in the video and recovering information regarding their 3D motion. Next come capabilities for capturing information about the shape of the 3D world. Clearly, it is important to have knowledge of the camera's motion and independently moving objects before shape information is recovered. Next come the appearance-based capabilities that recover the reflectance characteristics of the surfaces developed by the shape capabilities. Finally, at the top of the hierarchy reside the space capabilities which are responsible for creating data structures on which information about the 3D world extracted from the previous processes is organized with the goal of developing representations of the 3D world at various levels of abstraction.

It should be emphasized that video handling can draw a lot from the recently emerging paradigm of Qualitative Vision. In order to speak of an algorithm as

qualitative, the primitives to be computed do not have to rely on explicit unstable, quantitative models. Qualitativeness can be achieved in a number of ways: The primitives might be expressible in qualitative terms, or their computation might be derived from inexact measurements and pattern recognition techniques, or the computational model might be proved stable and robust in all possible cases. The lack of explicit camera information (focal length, center of visual field, other intrinsic parameters) in the video handling domain makes the use of qualitative techniques an absolute must. Our representations for motion, shape, appearance and space cannot be metric in the sense of encoding abstractions between objects in the 3D world as in the traditional approach pursued in Computer Image Analysis.

In the sequel, we describe a number of developments regarding motion-based and shape-based capabilities for video handling.

3.1 Motion-based Capabilities

Most current motion understanding techniques require the computation of exact image motion (optical flow in the differential case or correspondence of features in the discrete case). This is an ill-posed problem however, and additional assumptions about the 3D world and video formation process have to be employed. As a result, in the general case, the computed image displacements are imperfect. In turn, the recovery of 3D motion from noisy flow fields has turned out to be a problem of extreme sensitivity with small perturbations in the input causing large amounts of error in the motion parameter estimates. To overcome this problem, in our approach to the development of motion related capabilities, we skip the first computational step. The techniques developed [12] are based only on spatiotemporal derivatives of the video intensity function—the so-called normal flow. In fact, part of the analysis uses only the sign of the normal flow. It should be mentioned that a few techniques using normal flow have appeared in the literature; however, they deal only with restricted cases (only translation or only rotation [13, 14]).

A further characteristic is that the constraints developed for the motion-based capabilities, for which the rigid motion model is correct globally, are such that the video data is also globally utilized. The bases for these computations are global constraints which relate the spatial-temporal derivatives of the video intensity function globally to the camera motion parameters.

These global constraints are defined on the normal flow vectors. Given a normal flow field, the vectors are classified according to their directions. The vectors of each class have a certain structure that takes the form of patterns in the image (on the sphere or in the plane). For example, one can select in the plane normal flow vectors whose direction is defined with regard to a point with coordinates (r, s). These so-called copoint vectors (r, s) are the vectors which are perpendicular to straight lines passing through the point (r, s). In addition, the normal flow vectors of a class are distinguished as to whether their direction is counter-clockwise or clockwise with respect to (r, s), in which case they are called positive or negative (see Figure 1). Since any point (r, s) in the image can be

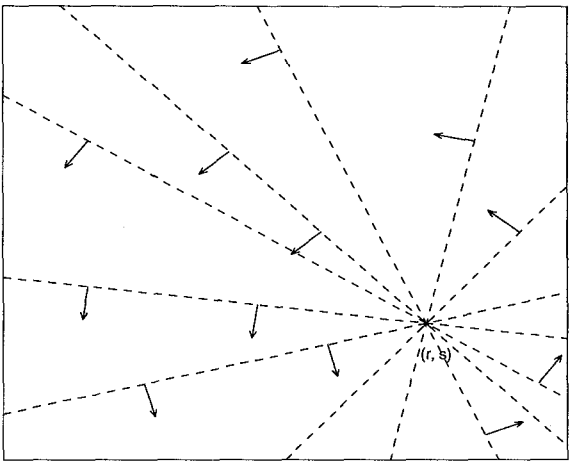


Fig. 1. Positive (r, s) copoint vectors.

chosen as a reference point, there exists an infinite number of such classifications.

Every class of copoint vectors has the following property: Considering only translational vectors, we find that the positive and negative vectors are separated by a line. In one half-plane the vectors are positive, in the other the vectors are negative, and on the line they are zero (Figure 2a). Vectors due to rotation, on the other hand, are separated by a conic section into positive and negative ones (Figure 2b). Vectors of general rigid motion (rotation and translation) thus obey the structure shown in Figure 2c. In one area the vectors are positive, in a second they are negative, and the vectors in the third area can take any value. This structure on the normal flow vectors is called the copoint pattern. Similar patterns exits for other classifications.

These findings allow us to formulate the problem of egomotion estimation as a pattern recognition problem. By localizing for different classes of normal flow vectors the positive and negative areas in the image plane, the parameter for the axis of translation and the direction of rotation can be derived [17].

Furthermore, based on the same basic constraints, the capability of the detection of independent motion has been designed. If the video capturing camera is stationary, detecting and localizing moving objects is easy. If the camera is moving, an area with a motion field not possibly due to one rigid motion must contain an independently moving object. The constraints are defined for the whole visual field, but also the motion vectors in every part of the image plane must obey a certain structure. Our approach consists of comparing the motion field within image patches with pre-stored patterns (which represent all possible rigid motions).

Thus, the output of the motion-based capabilities is a representation of egomotion and independent motion(s). Further capabilities, not discussed here, would be responsible for labeling the independent motion(s) with 3D motion

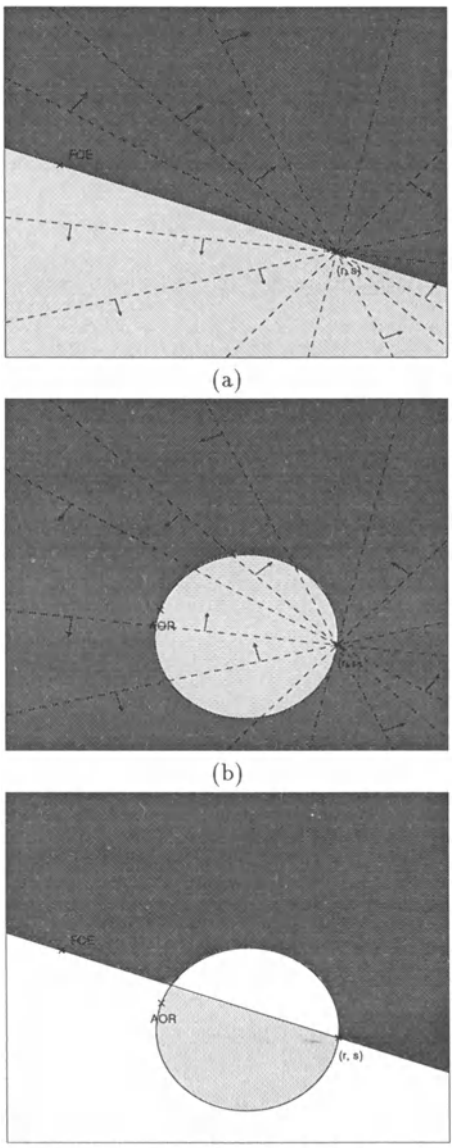


Fig. 2. (a) The translational (r, s) copoint vectors are separated by a line that passes through the FOE (the point which denotes the direction of translation); in one half-plane all vectors have positive values (light grey), in the other half-plane negative values (dark grey). (b) The rotational (r, s) copoint vectors are separated by a second order curve that passes through the AOR (the point where the rotation axis pierces the image plane). (c) A general rigid motion separates the (r, s) copoint vectors into an area of negative vectors, an area of positive vectors, and an area that may contain vectors of any value (white).

information (e.g., moving forward, left, right, fast, slow, characteristics of non-rigidity, periodicity, etc.)

3.2 Shape-based Capabilities

One of the goals of Computer Vision is the reconstruction of 3D descriptions of the world. A lot of effort has been devoted to developing techniques for computing metric shape and depth descriptions from 2D imagery. Studies concerned with this kind of work are collectively referred to as "shape from X" computations, where X is a cue such as shading, texture, pattern, motion, or stereo. Especially in the video handling domain it is unrealistic to expect that quantitative 3D structure can be computed. In most cases, metric depth is even impossible to compute since the intrinsic camera geometry is unknown.

Instead of computing metric structure, we could estimate qualitative information about the structure of the 3D world. This is possible given the results of motion-based capabilities. The traditional shape representation is the depth of features in the 3D world in view, or its derivatives such as shape, curvature, etc. We argue that the shape representations to be recovered by a video handling system are different from the metric ones traditionally used in the sense that they are not a 2 1/2-D sketch or functions thereof. We call such structures "qualitative structures," this basically amounts to recovering a function of the depth of the 3D world, with partial information about this function available. More specifically, if $z(x, y)$ is the function providing the depth z at points (x, y) in the image plane, and if $f_i, i = 1, 2, ..., n$ is a set of functions not necessarily exactly know, then $f_i(z(x, y)), i = 1, ..., n$ is a set of qualitative structures of the scene. For example, if f is a monotonic function in z with the property $z(x, y) < z(x', y') \Leftrightarrow f(z(x, y)) < f(z(x', y'))$, the f constitutes a qualitative representation that we call "ordinal structure," since knowledge about it only allows for ordering the values of the 3D world's depth. Functions with different properties amount to qualitative representations of a different nature. [18] contains a detailed description of ordinal shape computation.

4 Summary and Conclusions

We have described part of a hierarchy of representations that can be extracted from video sequences for the purpose of performing operations such as video retrieval, indexing, browsing, compressing and editing. The purpose of this paper was to set the foundations for a systematic study of the representations that can be robustly computed from video sequences which are independent of the particular type of video under consideration. That is, the video computations are purely bottom-up and domain specific knowledge about the video type is not used. In a future paper, we will address the problem of utilizing these representations for solving particular video handling problems. We will also develop dual representations at the level of the user interface of the video handling system and the mapping of these dual representations into video representations.

The representations we proposed form a natural hierarchy ranging from motion-based ones through shape-based and appearance-based representations, to representations of space. We described the first two in some detail and left out the appearance and space-based ones. In a nutshell, space representations emerge from integrating lower-level representations using a process resembling mosaicking performed on the level of ordinal shape descriptions coupled with appearance-based ones. The power of the approach lies in the fact that video representations are not embedded in a metric 3D Euclidian space. Rather, these representations are "qualitative" in the sense that they capture less information than the representations that Computer Vision algorithms traditionally attempt to compute. This, in turn, guarantees that the algorithms are robust with a well-defined output irrespective of the input video. We end by pointing out that the field of video processing can benefit a great deal from interaction with computer vision techniques leading to robust algorithms and methods which do not rely on assumptions about the structure of the scene.

References

1. Anastassiou, D.: Digital television. Proc. IEEE **82** (1994) 510–519
2. Tonomura, Y., Abe, S.: Content oriented visual interface using video icons for visual database systems. Journal of Visual Languages and Computing **1** (1990) 183–198
3. Ueda, H., Miyatake, T., Yoshizawa, S.: Impact: An interactive natural-motion-picture dedicated multimedia authoring system. Proc. Human Factors and Computing (1991) 343–350
4. Otsuji, K., Tonomura, Y.: projection detecting filter for video cut detection. ACM Multimedia (1993) 251–257
5. Teodosio, L.A., Bender, W.: Salient video stills: Content and context preserved. ACM Int'l. Conf. on Multimedia (1993)
6. Tonormura, Y., et al.: VideoMap and VideoSpaceIcon: Tools for anatomizing video content. ACM INTERCHI (1993) 131-136
7. Mann, S. Picard, R.W.: Virtual bellows: Constructing high quality stills from video. ICIP (1994)
8. Szeliski, R.: Image mosaicking for tele-reality applications. Technical Report CRL 94/2, DEC Cambridge Research Lab. (1994)
9. Sawhney, H.S., Ayer, S., Gorkani, M.: Model-based 2D & 3D dominant motion estimation for mosaicking and video representation. IBM Almaden Tech. Report (1994)
10. Niblack, W., et al: The QBIC project: Querying images by content using color, texture, and shape. SPIE 1908, Storage and Retrieval for Image and Video Databases (1993)
11. Jayant, N., Johnston, J., Safranek, R.: Signal compression based on models of human perception. Proc. IEEE **81** (1993) 1385–1422
12. Fermüller, C., Aloimonos, Y.: Qualitative egomotion. Int'l. Journal of Computer Vision (in press) (also available as a technical report from the Computer Vision Laboratory, Center for Automation Research, University of Maryland)

13. Aloimonos, J., Brown, C.: Direct processing of curvilinear sensor motion from a sequence of perspective image. Proc. Workshop on Computer Vision: Representation and Control (1984) 72–77
14. Horn, B., Weldon, E.: Computationally efficient methods for recovering translational motion. Proc. Int'l. Conference on Computer Vision (1987) 2–11
15. Fermüller, C., Aloimonos, Y.: The synthesis of vision and action. Exploratory Vision: The Active Eye (M.S. Landy, Ed.) Springer Verlag (1995)
16. Fermüller, C., Aloimonos, Y.: The geometry of visual correspondence. Proc. 5th Int'l. Conference on Computer Vision, Cambridge, Massachusetts (1995)
17. Fermüller, C.: Navigational preliminaries. Active Perception, Advances in Computer Vision (Y. Aloimonos, Ed.) Hillsdale, New Jersey: Lawrence Erlbaum Associates (1993)
18. Fermüller, C. Aloimonos, Y.: Representations for active vision. Proc. Int'l. Joint Conference on Artificial Intelligence, Montreal, Canada (1995) 20–26

Analysis of Left Ventricular Motion

Wen-Hung Liao, S. J. Aggarwal and J. K. Aggarwal *

Computer and Vision Research Center
Dept. of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712-1084

Abstract. This paper presents recent developments in medical imaging concerning the analysis of non-rigid heart motion. We first describe various imaging modalities that have been used to study the deformation of the left ventricle, including ultrasonic, magnetic resonance images(MRI), and computed tomography(CT). Three general approaches in analyzing the cardiac motion, namely, 1) artificial intelligence (AI)-based methods, 2) parametric models, and 3) physically-based analysis, are then identified and discussed. Finally, we present our research regarding the reconstruction of dynamic structure of the frog's ventricle imaged through a stereo light microscope(SLM).

1 Introduction

Characterization of myocardial motion enables the understanding of the physiology of the normal heart and detection of cardiovascular diseases. In clinical applications of cardiac imaging, the detection and tracking of the left ventricle(LV) boundary has traditionally been done through intensive human-machine interaction, which is time-consuming and often cannot be reliably reproduced. The need for automating the quantitative assessment of the non-rigid deformation of the left ventricle is apparent, and has led to the development of various image analysis techniques specially tuned for such applications.

The task of segmenting and tracking the LV boundary is a very challenging one due to the fact that the heart is a non-rigid object that rotates, translates and deforms in 3-D space. Motion analysis algorithms that rely on the rigidity assumption, therefore, can not be effectively applied to recover the deformation parameters. Relaxing the rigidity constraint, however, greatly increases the complexity of the tracking problem, as the degree of freedom of the motion is increased drastically. In particular, the heart motion belongs to the general class of non-rigid motion known as *elastic motion*, whose only constraint is some degree of continuity or smoothness. In this domain of deformable motion, there are no constraints other than topological invariance or coherence of bodies. A natural approach to deal with this type of motion is to assume an object model, and then try to model the deformations as variations to the model parameters.

* This work was supported by National Science Foundation, Contract BIR-9106624.

Earlier investigations on LV motion used invasive methods such as the surgical implantation of markers, followed by manual registration of the markers to reveal the complex structure and motion patterns. Recently, many non-invasive imaging modalities have been developed to capture the LV image the over the entire cardiac cycle, including ultrasonic images, magnetic resonance images(MRI), and computed tomography(CT). Depending on the characteristics of the acquired images, there exist different levels of difficulty associated with the tasks that need to be accomplished. The following list summarizes the objectives in the order of increasing complexity:

- 2-D boundary detection/segmentation.
- 2-D contour tracking.
- Stereo reconstruction and flow vector recovery.
- 3-D surface detection/segmentation.
- Volume modeling and tracking.

This paper aims to give a general overview on recent developments in medical imaging concerning the analysis of non-rigid heart motion. In section 2, we briefly review some non-invasive imaging techniques for data acquisition, along with the characteristics of acquired images. Section 3 presents the three general approaches in analyzing the cardiac motion, namely, 1) artificial intelligence(AI)-based methods, 2) parametric models, and 3) physically-based analysis. Section 4 reports the research we have done regarding the reconstruction of the dynamic 3-D structure of frog's ventricles using stereo microscope images. Finally, our conclusions on the current state of non-rigid heart motion analysis are presented in section 5.

2 Imaging Modalities

A simple anatomical model of the heart is depicted in Fig. 1. Much of the heart consists of a special type of muscle called myocardium. The internal surface of the heart is lined with a smooth membrane called endocardium, while the outer layer of the left ventricle is known as epicardium. A typical short axis slice of the heart (Fig. 2) reveals the relative position of the left ventricle, right ventricle, endocardium, epicardium, and myocardium. There are several imaging techniques that provide images of the heart and its structure, and they are briefly described in the following.

2.1 Ultrasonic Images

Echocardiography is a method of obtaining an image of the structure of the heart using *ultrasound*(inaudible, high frequency sound waves). The sound is reflected differently by each part of the heart, and a complex series of echoes results. Combining the different views produces the overall structure of the heart. Echocardiography is a major diagnostic tool used to detect the structural and functional abnormalities of the heart walls.

2.2 Computed Tomography

Cine-computed tomography (cine-CT) is a fast imaging modality capable of providing high quality cross-sectional image of the heart. Parallel two-dimensional slices of cine-CT can be stacked to form three-dimensional images. One major problem with cine-CT images is that they contain a lot of objects that are not of interest, since the entire chest is imaged. Automatic detection of the heart boundary is thus difficult due to the presence of multiple objects.

2.3 Magnetic Resonance Images

Magnetic resonance imaging is a diagnostic technique that provides high-quality cross-sectional images of organs and structures within the body without X rays or other radiation. It is the most commonly used imaging modality in studying the motion of the left ventricle. MR imaging with SPAtial Modulation of Magnetization (SPAMM) is a technique in which regions of the heart wall are tagged that their motion can be tracked in subsequent images. The tagging is done by producing localized regions of altered magnetization. The generation of magnetic saturation will result in the appearance of a dark band. The altered magnetization is a property of the tissue and will move with it. Therefore, the tissue motion can be inferred from the movement of the dark bands.

3 Image Analysis Methods

Numerous semi-automatic and automatic procedures for characterizing cardiac motion and locating heart boundary have been proposed during the last decade. These methods are built upon ideas ranging from AI-based procedures such as graph searching and fuzzy reasoning to physically motivated processes such as deformable models.

Earlier approaches tried to determine the boundary of the heart wall by standard edge detection schemes, followed by some pixel labeling procedure. Though simple and intuitive, they tend to generate boundaries that are not connected because of poor contrast and noise in the input data. Applying *a priori* knowledge such as shape smoothness and approximate location improves the results to some extent. However, methods based solely on local intensity information are, in general, insufficient to accurately describe the heart boundary. Such deficiency has motivated another class of approach known as global parametric models. By incorporating parametric models, the seemly unstructured elastic deformation can be compactly represented by a small number of parameters. They offer two distinctive advantages: 1) *a priori* knowledge can be implicitly integrated with the parametric model, and 2) the task of motion recovery is reduced to the problem of parameter estimation. Physically-based methods extend the parametric models further by augmenting them with the laws of rigid and non-rigid dynamics. They are especially useful for analyzing the dynamic behavior and modeling complex local deformations.

In the subsequent discussions, we review recent work in the three general areas identified above. In addition to describing the image sources, and the method being adopted to analyze them, we also explain how domain-specific information about the left ventricle is incorporated to increase the robustness of implemented algorithms.

3.1 AI-based Approaches

Suh *et al.* [33] developed a knowledge-based system for detecting the interior and external boundaries of the left ventricle from ECG-gated magnetic resonance images. Their system consists of three processing stages. The *low-level* processing performs a cartesian-to-polar coordinate conversion so that circular features such as the heart boundaries are transformed to vertical line segments. Candidate edge pixels are then detected by using a directional derivative operator. This is followed by *intermediate-level* processing in which databases are dynamically constructed from measurements made on each line segment. Finally, the line segments are labeled during the *high-level* processing phase through the Dempster-Schafer approximate reasoning theory - a framework for representing and combining multiple uncertain and imprecise. evidences from the knowledge databases. *A priori* knowledge about the relative positions of LV, RV, lung and blood pool is employed to build the knowledge source.

Weng *et al.* [37] developed another learning-based method to automatically detect ventricle boundary from cardiac MR and CT images. Their objective is to eliminate the difficult choice of proper thresholds that is usually encountered in the segmentation problem. The learning takes place in two stages: 1) learning for automatic selection of thresholds, and 2) learning for automatic selection of the region of interest from a number of candidate regions. During the learning phase, human interaction is required to define the knowledge base. After the knowledge database has been established, the system can then perform fully automatic boundary detection. No *a priori* knowledge about the heart shape is used, but their approach does assume that the region of interest is approximately homogeneous.

Thedens *et al.* [36] have applied the concept of graph searching for border detection in cardiac magnetic resonance image sequences. The formulation of edge detection as a graph searching problem is not new. However, most approaches use information from a single image, and they usually suffer from sensitivity to noise and occlusion. Thedens *et al.* extended the single frame graph searching method by considering a temporal or spatial sequence of images. They assume that the incorporation of the redundant information provided by a sequence of related image will increase the robustness of the resulting algorithm. Toward this end, they formulated the 3-D surface detection problem as a 2-D graph searching problem and developed heuristic search techniques to efficiently search for the (sub)-optimal path. Application of their algorithms to detect epicardial and endocardial surfaces has shown certain success.

Boudraa *at al.* [2] devised a method that uses the fuzzy ISODATA algorithm and Fourier analysis to detect the left ventricle contours. Fourier analysis is

performed on the image sequence to obtain the phase parameter that provides the temporal behavior of each pixel. The fuzzy ISODATA algorithm is essentially a classifier that groups data according to the local least mean square criterion. Applying the algorithm to the cardiac phase image will generate a given number of fuzzy subsets, in which LV forms a cluster. In this sense, the detection of LV boundary is reduced to the clustering problem. No assumption about the heart shape is required to guide the detection process.

Coppini *et al.* [8] proposed to reconstruct the 3-D shape of the left ventricle from echocardiographic (ultrasonic) images, based on a careful integration of existing computer vision algorithms with artificial neural systems. Their method can be decomposed into two parts: boundary detection and surface reconstruction. They have relied on the following physical assumptions about the LV surface: it is smooth, closed, and it does not fold. The boundary detection includes three phases: 1) edge detection using Laplacian-of-Gaussian (LoG) operator, 2) edge grouping by a linking process based on edge connectedness and, 3) classification of edge-segments using a two-layer feed forward neural network. The surface reconstruction process combines information from different views using a elastic model attached with springs. An objective functional consisting of the energy of the surface and the energy of the springs is formulated. A feed-forward neural network is then employed to minimize the energy functional to obtain the LV-surface shape. They have been able to reconstruct the dynamic shape of the LV using only four apical views of the heart.

Guttman *et al.*[11] presented methods for tag tracking and contour detection in tagged MR images. Tags were tracked using template matching, where the template is derived from a mathematical model of the MR tagging process. Inner and outer left ventricular boundaries are detected using the dynamic programming paradigm. Essentially, candidate contours points are extracted near the heart boundaries. (The user has to supply an approximate region of interest initially.) True edge points are then selected from the set of candidate points using a discrete multistage decision process, which can be solved efficiently using dynamic programming. To avoid erroneous results under poor imaging conditions, manual correction is allowed through user interaction with a GUI.

3.2 Parametric Models

A number of parametric models have been proposed for representing the LV contour and surfaces, including ellipsoid, superquadrics, spherical harmonics, rational gaussian curves and surface, and deformable Fourier descriptors. In the following, we briefly discuss these representation schemes and comment on their usage and limitations in modeling the non-rigid deformation of the heart.

Ellipsoid The shape of the heart immediately suggests a simple geometrical model using ellipsoids. An n-dimensional, origin-centered nondegenerate ellipsoid can be represented as follows:

$$\{z | z^T X^{-1} z = 1, z \in R^n\} \tag{1}$$

where X is a symmetric, positive definite $n \times n$ matrix that defines the ellipsoid. For example, an ellipse ($n = 2$) centered at the origin with semiaxis lengths a and b, and a rotation angle ϕ is characterized by the matrix:

$$X = R^T \begin{pmatrix} a^2 & 0 \\ 0 & b^2 \end{pmatrix} R. \tag{2}$$

where R is a 2×2 matrix:

$$R = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix}.$$

The above representation makes it easy to capture a wide range of ellipsoid dynamics, such as scaling, rotation and eccentricity change, through the following evolution equation:

$$X(k+1) = A(k)^T X(k) A(k) \tag{3}$$

where $A(k)$ models the shape deformation. Jaggi *et al.* [12] adopted the above formulation and modeled the left ventricle as a dynamically evolving 3-D ellipsoid. The actual noisy observation, $Y_i(k)$, are projections of the 3-D ellipsoid into 2-D ellipse, corrupted with noise $V_i(k)$:

$$Y_i(k) = C_i^T X(k) C_i + V_i(k) \tag{4}$$

The dynamic ellipsoid reconstruction problem is then casted into standard state-space form, which is subsequently solved using recursive estimation theory. Because their approach deals with ellipsoids, not images, a preprocessing stage is required to extract ellipses from the images—a non-trivial task when the image quality is poor.

Superquadrics Superquadrics are a family of parametric shapes derived from the parametric forms of the quadric surfaces. Mathematically, the equation for a superquadric surface is given as:

$$e(\eta, \omega) = \begin{bmatrix} a_1 C^{\epsilon_1}(\eta) C^{\epsilon_2}(\omega) \\ a_2 C^{\epsilon_1}(\eta) S^{\epsilon_2}(\omega) \\ a_3 S^{\epsilon_1}(\eta) \end{bmatrix}, \frac{-\pi}{2} \leq \eta \leq \frac{\pi}{2}, -\pi \leq \omega \leq \pi. \tag{5}$$

where

$$C^{\epsilon}(\eta) = sgn(\cos\eta)|\cos(\eta)|^{\epsilon},$$

and

$$S^{\epsilon}(\eta) = sgn(\sin\eta)|\sin(\eta)|^{\epsilon}.$$

The variables η and ω correspond to latitude and longitude in a spherical coordinate system. ϵ_1, ϵ_2 are known as *squareness* parameters and a_1, a_2, and a_3 are scaling factors. By varying these five parameters, a variety of shapes can be obtained. Global deformations (bending, tapering, twisting, and concavity) and local deformations can also be incorporated to enhance the modeling capabilities of superquadric surfaces[19].

Chen *et al.* [5, 4] have used superquadrics to model the global deformation of the left ventricle during the cardiac cycle. Bardinet *et al.* [1] have taken a similar approach. To accurately capture complex local shape deformations, however, more sophisticated modeling primitives are required. Chen *et al.* have chosen *spherical harmonics* to model the local variations, while Bardinet *et al.* augment the superquadric model with free-form deformations.

Rational gaussian curves/surfaces A rational gaussian curve is defined by:

$$P(u) = \sum_{i=1}^{n} V_i g_i(u), u \in [0,1] \tag{6}$$

where $g_i(u)$ is a rational function:

$$g_i(u) = \frac{w_i G_i(u)}{\sum_{j=1}^{n} w_j G_j(u)} \tag{7}$$

and

$$G_i(u) = \sum_{j=-J}^{J} \exp \frac{-[u-(u_i+j)]^2}{2\sigma_i^2} \tag{8}$$

J is a small integer ($0 \leq J \leq 5$). A rational gaussian surface can be obtained by substituting $G_i(u)$ with a two-dimensional gaussian function. In the above equations, n is the number of control vertices,V_i is the value of i-th control vertex, w_i is the weight associated with the i-th control vertex, and σ_i is the standard deviation of the gaussian function. Therefore, given a set of n points on the LV boundary, a smooth curve can be derived using the rational gaussian interpolation. Similarly, segmentation of 3-D images can be achieved using rational gaussian surfaces.

Goshtasby and Turner [10] have devised a two-stage algorithm to extract ventricular chambers in MR images. The first stage locates the approximate boundary of the LV using local gradient magnitude information. The second stage fits the approximate boundary using rational gaussian curves and surfaces. In our opinion, this approach is more useful in *modeling* and *animating* than in *analyzing* the ventricle wall motion.

Deformable Fourier descriptor Staib and Duncan [31] proposed a parametrically deformable model for boundary finding based on the elliptic Fourier decomposition of a contour :

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix} \tag{9}$$

The parameter vector to be recovered is:

$$\mathbf{p} = (a_0, c_0, a_1, b_1, c_1, d_1, ...).$$

The strength of their model stems from the combination of flexible constraints in the form of probabilistic models. Boundary detection is formulated as an estimation problem using a Maximum *A Posteriori* (MAP) objective function. LV boundaries in a sequence of MR images have been successfully identified with their method. In [32], the 2-D model is extended to allow global shape parametrization of smoothly deformable 3-D objects. Surface detection is formulated as an optimization problem, although no probabilistic model is involved. More recently, Chakraborty *et al.* [3] enhanced the deformable Fourier boundary descriptor by incorporating region based segmentation.

3.3 Physically-based Approaches

As we have mentioned previously, parametric models are sometimes inadequate for the analysis and representation of dynamic shape variation of the left ventricle. To expand the modeling power and make better use of the physics property of the LV, a number of physically-based non-rigid motion analysis methods have been developed. The following discussion will concentrate on two important physically-motivated models: *snakes* and modal analysis.

3.4 Snake and its derivatives

The *snake* model is perhaps the most well-known and widely used physically-based model. A great deal of attention has been devoted to the extension of the named prototype first proposed by Kass *et al.* [13]. The original snake model is a class of active contours that evolve under the influence of external potentials but are constrained by internal energies. To be exact, let $v(s) = (x(s), y(s))$ be the parametric representation of the position of a curve, then the energy functional for a snake can be expressed as:

$$E_{snake} = \int_0^1 E_{snake}(v(s))ds \tag{10}$$

where E_{snake} is a composition of three energy terms: 1) E_{int}: internal spline energy due to stretching and bending, 2) E_{image}: image energy, and 3) E_{con}: external constraint energy. In most cases, internal spline energy takes the following form:

$$E_{int} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2 \tag{11}$$

The spline energy is composed of a first-order term controlled by $\alpha(s)$ and a second-order term controlled by $\beta(s)$. The parameters $\alpha(s)$ and $\beta(s)$ account for the material's resistance to stretching and bending, respectively. The image energy varies according to the application. In most LV boundary detection problem, edge functional is adopted.

Dynamic snakes with intuitive physical behaviors are developed [35] when the original model is augmented with Lagrangian mechanics. In [9], temporal context is explicitly incorporated to form a spatio-temporal solid called the *active tube*. Not surprising, the snake model can also be generalized to deal with 3-D

images, as shown by Cohen *et al.* in [6, 7]. This newly formed class of deformable surfaces, called *balloons*, is able to conform to image features and external forces in a way similar to the original snake model. An evolution equation similar to that in dynamic *snakes* has also been formulated. Cohen *et al.* have successfully applied this model to the segmentation of 3-D MRI images as well as to establishing correspondence between a deformable surface and an anatomical atlas. McInerney *et al.* [18] have addressed similar problems using a slightly different balloon model. Their focus, however, is on the numerical solution of the dynamic deformable surface model.

A more recent extension to *snake* models can be found in [34], where the essential elements of physically-based models and probabilistic approaches are incorporated. A Bayesian framework is introduced and the original energy-minimizing problem is transformed to an MAP problem. To further exploit the power of probabilistic modeling, Szeliski *et al.* [34] have developed a sequential estimation algorithm using the Kalman filter. Known as the *Kalman snake*, this dynamic system is able to integrate non-stationary, noisy observations over time. It provides the flexibility to design behaviors that may not be possible with pure physically-based models. Moreover, model parameters can be derived from statistical models of sensors, rather than chosen heuristically. It would be useful to generalize their development to form *Kalman balloons* that are capable of tracking dynamic surfaces.

A significant amount of work in LV boundary detection/tracking has been influenced by the *snake* model. For example, Ranganath [26] developed an automatic contour extraction procedure using a specially configured active contour model. Contour propagation between image slices (spatial continuity) and image phases (temporal continuity) is considered to enhance the reliability of the extracted boundaries.

Kumar and Goldgof [15] also based their feature tracking algorithm on snake models. Spatio-temporal tracking of the SPAMM grid in cardiac MR images is achieved by using multiple parallel and vertical snakes. The intersections of snakes are used as markers for establishing point correspondences. After the correspondence problem is solved, a thin-plate spline model is utilized to interpolate the local motion vectors.

Along a similar vein, Young *et al.* [39, 38] proposed deformable models to reconstruct 2-D and 3-D heart wall motion from tagged MR images. Their approach appears to be very similar to that reported by Kumar and Goldgof, except that the latter used modified internal and image energies, while Young *et al.* used the original snake formulation and minimization procedure developed by Kass *et al.* Finite element method is employed to model the geometry and deformation of the LV. Verification of the proposed algorithms is achieved using a silicon gel phantom undergoing controlled deformation.

The semi-automatic tag tracking algorithm proposed by Kraitchman *et al.* [14] represents a combination of template matching and active contour modeling. Tag intersections are first detected by computing the normalized cross-correlation between an idealized template and the tagged MR image. An 'active

spring mesh' structure is then formulated to track the identified intersection points. The spring mesh has a different internal energy from the original snake model since discrete points, not lines or contours, are being tracked. The image energy is chosen to be the correlation image to attract the intersection points to regions of high correlation.

Lobregt and Viergever [17] presented a *discrete* dynamic contour model with special properties to avoid undesirable effects such as shrinking and vertex clustering, which are common problems in existing active contour models. The internal energy of the model depends on local contour curvature, while the external energy is derived from image features. The resulting model is both conceptually and computationally simple, and has proven to be useful for segmenting MI images. Extension of their methods to 3-D images, however, remains an open question.

3.5 Modal Analysis

Inspired by modal analysis in linear mechanical systems, Pentland [23] developed a system that is capable of automatically recovering deformable part models based on the *finite element method* (FEM). Non-rigid object behavior is described by *modal dynamics*, i.e., by the superposition of its natural strain or vibration modes. By limiting the number of modes used in the representation, the analysis of non-rigid motion can always be transformed to an overconstrained problem. Later, the same model combined with an extended Kalman filter is applied to recover non-rigid motion and structure from contour [25] as well as optical flow data [24]. A major limitation of the modal framework is that objects must be described in term of the modes of some prototype shape. Such a procedure implicitly imposes an *a priori* parameterization upon the sensor data. It is thus more suitable for *modeling* than for *tracking* purposes. To address this problem, Sclaroff and Pentland[27, 28, 29] recently developed a new method that computes the object's vibration modes directly from the image data. In [30], they applied this method to fit the X-ray contour of the left ventricle.

Nastar and Ayache [20] have attempted to unify the work of Terzopoulos *et al.* and Pentland *et al.* They followed similar physics principles and developed elastic models for non-rigid motion tracking. The notable property of their model is that the governing dynamic equations are linear and decoupled for each coordinate, regardless of the amplitude of deformation. Algorithmic complexity is therefore significantly reduced. In [22], they have applied this improved model to study the temporal evolution of 3-D non-rigid objects. In [21], they employed the physics-based vibration analysis to classify non-rigid heart motion in ultrasonic images as well as MRI data.

4 Shape Change Analysis of SLM Images

In this section, we briefly summarize our research concerning the reconstruction of the dynamic 3-D structure of a frog's ventricle imaged through a stereo light

microscope(SLM). (Please refer to [16] for details.) We have devised an integrated approach for the detection of heart boundary as well as the analysis of the shape deformation. Specifically, we have developed a motion-based segmentation process, augmented by an active contour model, for robust region-of-interest (ROI) extraction. Dynamic surface reconstruction is achieved by formulating spatio-temporal feature tracking as a four-dimensional graph matching problem.

4.1 Image Registration

Fig. 3 shows a stereo sequence of the frog's ventricle acquired by our SLM imaging system. The undesirable vertical shift between the images may be clearly observed, and is corrected by a correlation-based approach. The image registration algorithm involves two stages. First, a reliable template is selected in one image. (This is in contrast to conventional pattern matching where a template of interest is pre-specified.) Next, the corresponding feature in the other image is obtained by computing some likelihood measure. Once the correspondence is correctly established, the vertical shift is just the difference between the $y-$axis coordinates.

4.2 Motion-based Segmentation

Our segmentation procedure consists of the following steps. First, we apply spatio-temporal filtering to emphasize regions of large intensity variations in both the spatial and temporal domains. In the spatial domain, we use a Laplacian of Gaussian (LoG) filter, while in the temporal domain, we resort to a simple differencing technique due to the limitation in the temporal sampling rate. The resulting image is thresholded to obtain the approximate ROI. Normally, the approximate ROI contains holes and exhibits an irregular boundary. To smooth the object boundary, we apply morphological filtering (CLOSING and OPENING) with an approximately circular structuring element several times. Finally, a blob-coloring technique is utilized to extract the final ROI. It is noteworthy that the result of segmentation not only provides a boundary condition for stereo matching, but also constitutes the initial estimate for non-rigid motion estimation.

The approach discussed above has achieved success in segmenting the frog's ventricle. Still, the resulting boundary remains somewhat ragged. One can apply morphological filters to further smooth the boundary. But there is obviously a trade-off between the smoothness of the boundary curve and the accuracy of the object segmentation. To refine the boundary without excessively smoothing, we have used the *snake* model, with a modified energy functional, to segment and track the deformable contours over time. Iterative minimization of the snake energy calls for an initial guess that is not too distant from the true solution. Such an initial condition is readily available from the motion-based segmentation results presented earlier. We perform uniform sampling on the approximate contour and use these points as the initial input to the iterative estimation of the desired contour.

4.3 Cooperative Spatial and Temporal Matching Process

To recover the binocular image flows, we adopt a feature-based approach for establishing correspondence in both the spatial and temporal domains. The original correspondence problem, which involves only two sets of features, can be cast into a maximum weighted bipartite matching problem. This problem is known to be solvable in polynomial time by the Hungarian algorithm. However, applying the same principle to the situation when four distinct sets of features are concerned results in a much more complex, four-dimensional maximum weighted matching problem. Such problems belong to the class of *NP-complete* problems that currently permit no polynomial-time solution. In order to incorporate spatial and temporal constraints simultaneously, while maintaining the complexity of the resulting algorithm at a manageable level, we modify the original formulation and outline an algorithm that solves the modified problem in polynomial time. To be more specific, we construct relaxation labeling algorithms to effectively reduce the number of possible matches and then apply a local search algorithm to maximize the overall compatibility measure among the remaining correspondences.

A formal definition of the four-dimensional maximum weighted matching problem is given as follows:

INSTANCE: A set $M \subseteq W \times X \times Y \times Z$ where W, X, Y, and Z are disjoint sets having the same number of q elements and non-negative numbers w_{WX}, w_{XY}, w_{YZ} and w_{ZW} associated with the edges $[v_W, v_X]$, $[v_X, v_Y]$, $[v_Y, v_Z]$ and $[v_Z, v_W]$, respectively.

PROBLEM: Find a subset $M' \subset M$ such that 1) the cardinality of M', $|M'| = q$, 2) no two elements of of M' agree in any coordinate, and 3) the sum of the weights of edges connecting nodes in W, X, Y and Z is maximum.

Our objective is to find mutually compatible nodes in both spatial and temporal domains. It is a common situation that a feature does not appear in all images due to occlusion or deformation. In such cases, we just add null nodes with zero weights to fulfill the requirement that all sets have the same number of elements. The solution we seek is a *perfect* matching that maximizes the sum of connection weights. To get around NP-completeness, we use a relaxation labeling technique to reduce the problem size by considering domain-specific heuristics such as 'disparity smoothness constraint' in spatial matching and 'smoothness of motion constraint' in temporal matching. After the number of candidate matches is reduced, we apply a greedy-type local search algorithm to obtain the most compatible quadruple sets. We show an example of applying this cooperative matching process in Fig. 4.

Currently, we are exploiting domain-specific knowledge to enhance the performance of the matching algorithm. Plausible modifications include: (1) matching the features hierarchically, (2) using the extracted contour to guide temporal matching and (3) assigning more weights to favor the stereo matching results. The efficacy of the modified algorithms will be tested using a modal synthesized image sequence.

5 Conclusions

In this paper, we have surveyed recent advances in non-rigid left ventricular motion analysis. We have outlined three general approaches for segmenting and tracking the heart motion. Cardiac image analysis methods, however, are not necessarily limited to the categories we defined. With the advent of new imaging modality, we expect novel schemes will be developed to obtain more accurate quantitative assessments and functional evaluation of the left ventricle.

References

1. E. Bardinet, L. D. Cohen, and N. Ayache. Fitting of iso-surfaces using superquadrics and free-form deformations. *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pages 184–193, 1994.
2. A. E. Q. Boudraa, J. Malletand J. Bessenand S. Bouyoucef, and J. Champier. Left venrticle automated detection method in gated isotopic ventriculograhy using fuzzy clustering. *IEEE Trans. on Medical Imaging*, 12(3):451–465, 1993.
3. A. Chakraborty, L. H. Staib, and J. S. Duncan. An integrated approach to boundary finding in medical images. *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pages 13–22, 1994.
4. C. W. Chen, T. S. Huang, and M. Arrott. Modeling, analysis and visualization of left ventricle shape and motion by hierarchical decomposition. *IEEE Trans. on PAMI*, 16(4):342–356, 1994.
5. C.W. Chen and T.S. Huang. Surface modeling in heart motion analysis. *Proc. SPIE Vol. 1610: Curves and Surfaces in Computer Vision and Graphics*, pages 360–371, 1991.
6. I. Cohen, L. Cohen, and N. Ayache. Introducing new deformable to segment 3d images. *Proc. CVPR*, pages 738–739, 1991.
7. L. D. Cohen and I. Cohen. Deformable models for 3D medical images using finite elements and balloons. *Proceedings of International Conference on Pattern Recognition*, pages 592–598, 1992.
8. G. Coppini, R. Poli, and G. Valli. Recovery of 3-D shape of the left ventricle from echocardiographic images. *IEEE Trans. on Medical Imaging*, 14(2):301–317, 1995.
9. R. Furukawa, M. Imai, and T. Uno. Robust algorithm for motion analysis based on active tubes. *Proceedings of the Workshop on Motion of Non-rigid and Articulated Objects*, pages 200–205, 1994.
10. A. Goshtasby and D. A. Turner. Segmentation of cardiac cine MR images for extraction of right and left ventricular chambers. *IEEE Trans. on Medical Imaging*, 14(1):56–64, 1995.
11. M. A. Guttman, J. Prince, and E. R. McVeigh. Tag and contour detection in tagged MR images of the left ventricle. *IEEE Trans. on Medical Imaging*, 13(1):74–88, 1994.
12. S. Jaggi, W. C. Karl, and A. S. Willsky. Estimation of dynamically evolving ellipsoids with applications to medical images. *IEEE Trans. on Medical Imaging*, 14(2):249–258, 1995.
13. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):312–331, 1987.

14. D. L. Kraitchman, A. A. Young, C. N. Chang, and L. Axel. Semi-automatic tracking of myocardial motion in MR tagged images. *IEEE Transactions on Medical Imaging*, 14(3):422–433, 1995.
15. S. Kumar and D. Goldgof. Automatic tracking of SPAMM grid and the estimation of deformation parameters from cardiac mr images. *IEEE Trans. on Medical Imaging*, 13(1):122–133, 1994.
16. W. Liao, S. J. Aggarwal, and J. K. Aggarwal. Reconstruction of dynamic 3-D structures of biological objects using stereo microscopy. *Proceedings of 1st Int'l Conf. on Image Processing*, III:731–735, 1994.
17. S. Lobregt and M. A. Viergever. A discrete dybnamic contour model. *IEEE Trans. on Medical Imaging*, 14(1):12–24, 1995.
18. T. McInerney and D. Terzopoulos. A finite element method for 3D shape reconstruction and non-rigid motion tracking. *Proc.of 12th ICPR*, pages 518–523, 1993.
19. D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and non-rigid motion tracking. *Proc. CVPR*, pages 337–343, 1991.
20. C. Nastar and N. Ayache. A physically based analysis for deformations in 3D images. *Geometric Methods in Computer Vision II*, SPIE 2031:182–192, 1993.
21. C. Nastar and N. Ayache. Classification of non-rigid motion in 3D images using physics-based vibration analysis. *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pages 61–69, 1994.
22. C. Nastar and N. Ayache. Spatio-temporal analysis of nonrigid motion from 4D data. *Proceedings of the Workshop on Motion of Non-rigid and Articulated Objects*, pages 146–151, 1994.
23. A. Pentland. Automatic extraction of deformable part models. *IJCV*, 4:107–126, 1990.
24. A. Pentland and B. Horwitz. Recovery of non-rigid motion and structures. *IEEE Trans. on PAMI*, 13(7):730–742, 1991.
25. A. Pentland, B. Horwitz, and S. Sclaroff. Non-rigid motion and structure from contour. *Proc. CVPR*, pages 288–293, 1991.
26. S. Ranganath. Contour extraction from cardiac MRI studies using snakes. *IEEE Trans. on Medical Imaging*, 14(2):328–338, 1995.
27. S. Sclaroff and A. Pentland. A modal framework for correspondence and description. *Proc.of 12th ICPR*, pages 308–313, 1993.
28. S. Sclaroff and A. Pentland. Physically-based combinations of views: representing rigid and non-rigid motion. *Proceedings of the Workshop on Motion of Non-rigid and Articulated Objects*, pages 158–164, 1994.
29. S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Transcations on PAMI*, 17(6):545–561, 1995.
30. S. Sclaroff and A. P. Pentland. On modal matching for medical images: unconstrained shape description and data compression. *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pages 70–79, 1994.
31. L. H. Staib and J. S. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. on PAMI*, 14(11):1061–1075, 1992.
32. L. H. Staib and J. S. Duncan. Deformable fourier models for surface finding in 3D images. *Proc. SPIE Vol 1808: Visualization in Biomedical Computing*, pages 90–104, 1992.
33. D. Y. Suh, R. L. Eisner, R. M. Mersereau, and R. I. Pettigrew. Knowledge-based system for boundary detection of four-dimensional cardiac magnetic resonance image sequences. *IEEE Trans. on Medical Imaging*, 12(1):65–72, 1993.

34. R. Szeliski and D. Terzopoulos. Physically-based and probabilistic models for computer vision. *Proc. SPIE Vol 1570: Geometric Methods in Computer Vision*, pages 140–152, 1991.
35. D. Terzopoulos. On matching deformable models to images: Direct and iterative solutions. *Topical Meeting on Machine Vision, Technical Digest Series*, 12:160–167, 1987.
36. D. R. Thedens, D. Skorton, and S. Fleagle. Methods of graph searching for border detection in image sequences with applications to cardiac magnetic resonance imaging. *IEEE Trans. on Medical Imaging*, 14(1):42–55, 1995.
37. J. Weng, A. Singh, and M. Y. Chiu. Learning-based ventricle detection from cardiac MR and CT images. *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pages 23–32, 1994.
38. A. A. Young, D. L. Kraitchman, , L. Dougherty, and L. Axel. Tracking and finite element analysis of stripe deformation in magnetic resonance tagging. *IEEE Transcations on Medical Imaging*, 14(3):413–421, 1995.
39. A. A. Young, D. L. Kraitchman, and L. Axel. Deformable models for tagged MR images: reconstruction of two and three-dimensional heart wall motion. *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pages 317–323, 1994.

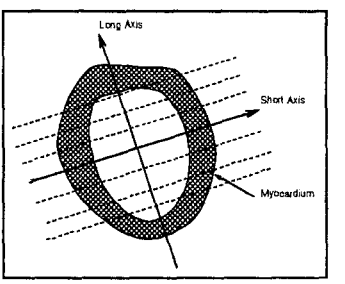


Fig. 1. A simple anatomical model of the heart.

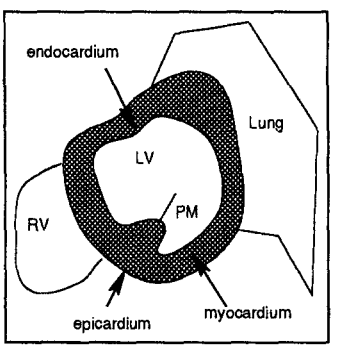


Fig. 2. The representative short axis slice shows: LV=left ventricle, RV=right ventricle, and PM=papillary muscle.

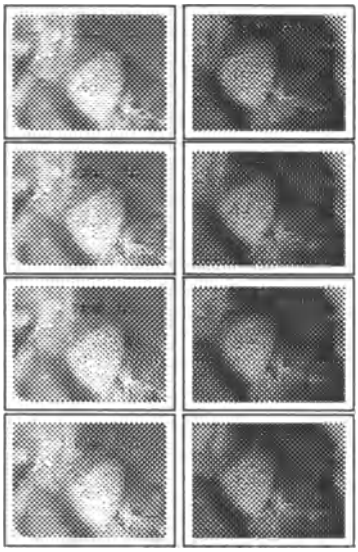


Fig. 3. Stereo image sequences of a frog's ventricle.

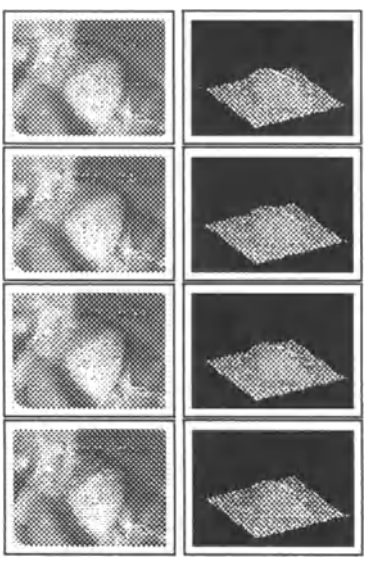


Fig. 4. (Left): Original image sequence,(Right): Reconstructed 3-D structure.

Applications of Computer Vision in Space Robotics

Chiun-Hong Chien

Hernandez Engineering Inc.
Houston, TX 77058
chien@superman.jsc.nasa.gov

Abstract. Automation using intelligent robotics has been recognized as a promising approach to earth-bound applications such as nuclear waste clean up, and factory automation. Successes in these applications, however, have not extended the applications of intelligent robotics to space systems. The advance of VLSI and sensor technologies, as well as maturity of research in computer vision make it feasible to exploit intelligent robots in space environment. This paper gives a list of target applications, discusses vision requirements, points out critical issues, and describes research activities conducted in applications of computer vision in space robotics. Issues addressed in those research activities include lighting, geometric calibration, sensor positioning, knowledge-based spatial reasoning, terrain mapping, vision-guided tele-operation, and real-time vision-guided grasping of free-floating objects, vision system planner, and computer architecture for real-time vision processing, among others. It is intended to encourage researchers in computer vision community to pursue great research opportunities and challenges in space robotics.

1 Introduction

Automation using intelligent robotics has been recognized as a promising approach to earth-bound applications such as nuclear waste clean up, and factory automation. Successes in these applications, however, have not extended the applications of intelligent robotics to space systems. In part, this has been due to scepticism regarding the viability of robotics technology and a strong concern for safety of flight hardware and personnel.

In order to evaluate the feasibility of robotics technology in space applications, one may need to understand the characteristics of space environment (including planet surface systems). Space environment has been characterized as hostile and extreme in a sense that it has large thermal gradients which cycle over time; it is full of radiation, meteroids, and debris; dust storm is common in planets such as Mars; the lack of atmospheric dispersion of the harsh solar light results in unfavorable illumination condition with a large dynamic range and sharp shadows. Furthermore, the remoteness of space results in constraints (e.g. light weight and low power) on and delayed access (e.g. several seconds to several minutes transmission delay) to equipment and resources (e.g. equipment and astronauts), and incomplete information (e.g. limited by resolution of orbital cameras).

The advance of VLSI technology makes it possible to pack much higher computation power and higher capacity storage into a smaller package, which in turns increase resolution of newly designed orbital cameras, and allow software engineers to design more sophisticated algorithms to deal special requirements for space applications of intelligent robotics. More importantly, maturity in computer vision research has earned itself an indispensable role in intelligent robotics (especially in a dynamic and unknown environment such as space and planet surface systems). RObot Technology EXperiment (ROTEX) conducted in German's Spacelab D2 on board Shuttle Columbia marked the first remotely controlled (by operators in a ground station, and by astronauts on Columbia with aid from enhanced graphics generated using stereo vision) robot in space [15], and opened the door for the applications of intelligent robotics in space. A space intelligent robot, known as ExtraVehicular Activity Helper and Retriever, under development at NASA Johnson Space Center has been scheduled to conduct flight experiments (i.e. real-time vision-guided grasping of free floating objects) on board NASA's KC-135 aircraft, and intended to extend experiments in space (e.g. Space Shuttle).

In this paper, we will give potential target applications of computer vision in space robotics, discuss vision requirements for these applications. Also included in the paper are brief descriptions of various space-related research on computer vision.

2 Target Applications

Various studies have been conducted to identify the types of robotics functional capability needed by planet surface systems [11], and space station. Those requires vision-assisted are listed below:

- autonomous precision landing on planet surfaces (e.g. Moon/Mars),
- surveying/building terrain maps of planet surfaces,
- loading and unloading equipment at planet outposts,
- regolith and rock handling (surface, hauling, mining, excavation),
- performing routine testing, inspection, and monitoring instruments,
- repair or replacement (e.g. satellite servicing, and Hubble a telescope repair), and parts inventory (storage and retrieval),
- sensing augmentation (extra lighting, measurements, placement, data patterns, ad-hoc images, etc.) and optimal light/sensor placement,
- preventive maintenance (contamination removal, tightening, etc.),
- setting up and dismantling work sites, and assembly and disassembly (e.g. space station truss assembly),
- prepare meal and harvest fruit or vegetable,
- deploying and retrieving payloads,
- EVA helpers (for tasks generally believed to be difficult for robotic systems to perform by themselves, including thermal blanket spreading, flexible shield emplacement, go-fer, etc).

3 Vision Requirements

In order to perform the afore-mentioned tasks, a set of visual perception requirements for space robotics have been proposed [12]. The requirements include,

- Be able to provide for every sensed attribute and object described below a measure of confidence in the inference of that attribute/object as a function of circumstance.
- In the context of providing continually sensed properties of the robot's environment, be able to selectively and recursively:
  - Focus attention to areas of interest in images.
  - Provide appropriate types of visual perception processing at the goal-directed request of the reasoning system.
  - Unify information derived from multiple sensors to increase confidence in visual perceptual inferences made.
  - Request modification of the perception environment to increase perceptual confidence as needed (by providing lighting/shade and repositioning sensors, etc.)
- Be able to perceptually find, recognize, spatially locate, and keep track of moving and stationary objects in world and moving-robot relative frames of reference in sparse and cluttered environments for periods longer than a few decision cycles.
- Be able to perceptually determine motion properties of objects in world and moving-robot relative frames from visual sensor data (translation and rotation parameters, time to collision, etc.)
- Be able to perceptually determine spatial relationships of objects from visual sensor data including robotic arms, hands, tools, and payloads, especially relative pose of objects and hands/tools in moving-robot relative frames.
- Be able to provide visual perception for vision-guided grasping of moving or stationary objects in world and moving-robot relative frames in hard (bounded) real time.
- Be able to build and continuously update useful representations of the local and global environment from perceptual processing of visual sensor data, including recognition of unmarked "distinctive places" for use by reasoning and planning.
- Be able to self-locate through visual perception in a local relative frame (within a global absolute frame) to an accuracy at a known time needed for task success.
- Be able to find objects in the current world which are specified by name or appropriate attribute subset and which require recall of prior perception episodes for use as a model for recognition and prior location.

4 Research Activities

Various research and development activities are being geared toward applications of computer vision in space robotics. This sections give brief descriptions of some of the related projects.

4.1 A Lighting System for Solar Illumination Simulation

In orbit, surface appearance can change drastically due to the variation in ambient light from solar and earth. The lack of atmospheric dispersion of the harsh solar light results in images having a large dynamic range with sharp shadows. To investigate the effect of solar illumination on space environment, simulation of solar illumination is currently under development by Jet Propulsion Laboratory (JPL) [23]. It is part of NASA sponsored *The Remote Surface Inspection Task* (RSI), a five year research project to investigate and develop ways to telerobotically inspect Space Station Freedom (SSF). A team at JPL have developed a small scale simulator which effectively mimics the relative motion of the Sun in the sky, while still providing realistically scaled illumination levels. The system is composed of a spotlight mounted on a pan/tilt/translate platform. Its ability to move spatially as well as modify its beam shape, enable it to provide constant intensity illumination that changes angle of incidence at rates equal to those experienced in low earth orbit.

4.2 Ground Servicing Operations on the Space Shuttle

Other projects under RSI include automated anomaly detection for High Temperature Reusable Surface Insulation (HRSI) of Thermal Protection System (TPS) conducted by NASA Langley Research Center, along with NASA Kennedy Space Center, Carnegie Mellon University and SRI international. The proposed anomaly detection system [8] compares a pair of images of each of Shuttle tiles to flag the anomaly for defect classification by a human inspector. The pair consists of an image taken after the Orbiter returns from a mission and a corresponding image previously recorded prior to flight. Image registration is required for aligning the two images before image comparison to accommodate small deviations in the camera viewpoints. A derive pattern vector procedure is used to produces a vector of four values representing a measure of similarity between the corresponding pre-fight and re-registered post-flight regions (i.e. tiles). This vector is then processed by linear classification and quadratic classification functions to discriminate actual anomalies from processing artifacts.

In order to alleviate illumination problems, ambient light subtraction used two image data set to obtain a compensated image. The first data set is illuminated only with the ambient light and the second is illuminated with the ambient light as well as the artificial illuminator (with strobe illumination to minimize the effect due to moving platforms).

4.3 A Knowledge-based Robotic Vision System

Space applications present environments in which it is desirable (and possible) to solve robotic problems using a model-based (or knowledge-based) approach.

Knowledge about target objects and the "engineered" environment allows expectation driven reasoning (as opposed to data driven unconstrained searches), which in turns makes it possible to exploit knowledge derived from features that have been found, thus expediting subsequent searches. A knowledge-based computer vision system has been developed for guiding a robot arm to perform tasks on a reconfigurable task panel [20]. The system was originally developed for an industrial environment, but can be applied to autonomous orbiter payload bay operations. The knowledge base consists of geometric and kinematic information and is utilized at several stages of spatial reasoning, including (a) worksite registration, (b) substructure (such as a hinged door of the reconfigurable task panel) generate and test, (c) camera repositioning (i.e. active vision) for higher accuracy and richer information, and (d) final manipulation of the substructure by the robot. The details of the system and the proposed spatial reasoning methodology can be found in [20].

4.4 Terrain Mapping

A six-legged robot, known as Ambler, has been built by Carnegie Mellon University for an exploratory mission on another planet, including Lunar and Mars, where it is to traverse unchartered areas and collect material samples. The first task for the (vision system) of Ambler is to construct an elevation map, a geometric terrain representation of rugged planetary surface using a 3D range sensor. A locus method is used to accurately convert a range image into an elevation map [17]. Incremental modeling of rugged terrain is then carried out by integrating multiple elevation maps from a sequence of range images using a pixel-based iconic matching method. To avoid error accumulation in a composite terrain map obtained over a long distance of robot motion, the locus method is exploited to compare the vehicle positions in both a prior digital elevation map (DEM) and the composite terrain map to reduce any potential errors. Their proposed 3D terrain mapping techniques have been used to built a composite terrain map by merging 125 real range images over a distance of 100 meters.

Autonomous navigation requires high-level scene descriptions as well as geometrical representation of the natural terrain environments. New algorithms have also been developed for extracting topographic features (i.e. peaks, pits, ravines, and ridges) from contour maps which can be generated from elevation map [17].

Sensors other than laser scanners have also been used to acquire information for terrain mapping. In [18], the stereo vision paradigm was utilized to yield an initial sparse range map, which was later combined with shape from shading in a data fusion module to produce a dense map. Finally the "current" terrain surface estimate was rotated to a topographic representation of the terrain that provides valuable data and facilitates path planning activities.

4.5 Vision Guided Tele-operation

ROTEX has been claimed as the first remotely controlled robot in space [15]. One of the experiments conducted in German's Spacelab D2 on board Shuttle Columbia was to grasp an object freely floating in space in a confined workcell after it was released (and thus initial object states were known). The relative position between the object and 6 DOF robot arm at any given time was to be determined from a camera in the hand of the robot using an approach (as described in [9]) exploiting the dynamical models for the object to be caught. The challenge was that the visual interpretation was performed at a ground station, and there was a significant delay (about 6.5 sec round trip) between the imaging time and the control signal reached to the robot in the Spacelab. This time-delay was compensated by using "remotely-brained machine vision" (a term coined by Dickmanns [9]).

4.6 Real-time Vision Guided Grasping of Moving Object

Another free flying robot, known as EVAHR (Extra-Vehicular Activity Helper and Retriever), is currently under development at Automation and Robotics Division of NASA Johnson Space Center. As its name implies, the EVAHR's primary responsibilities are to retrieve free flying objects, and perform EVAs in cooperation with crew members in the neighborhood of Space Station Freedom (or planetary habitat) (including tasks mentioned earlier).

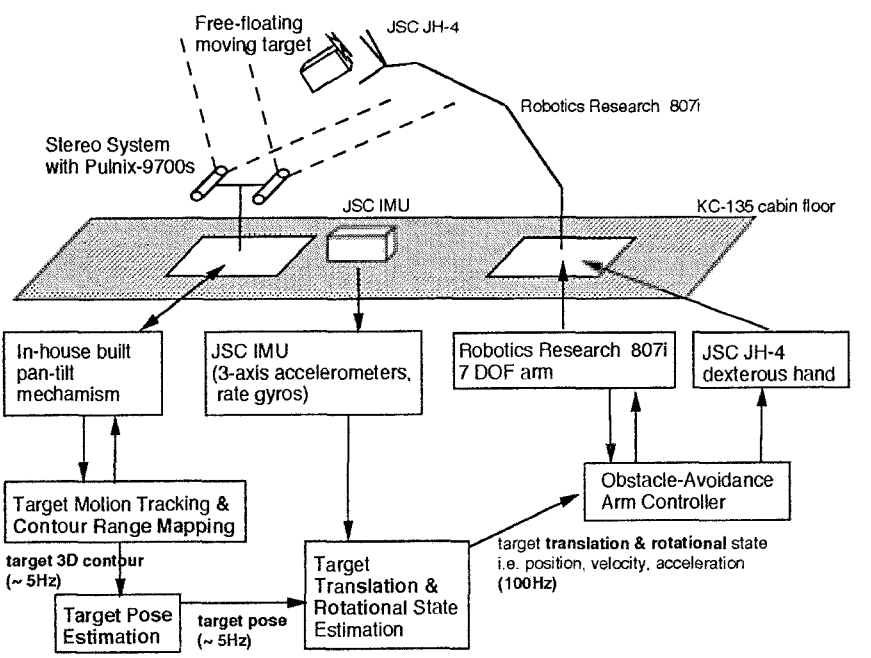


Figure 1: The EVAHR's software and hardware integrated system

The block diagram of the EVAHR software and hardware architecture is

shown in Figure 1. The system consists of an active vision system [6] [7], a state estimator [14], an arm/hand controller (responsible for action) [1], and an arm trajectory planner. All these modules are communicated with each other directly or through a world model storing various information (including recent history) about objects in the environment.

The EVHAR's vision system consists of a pair of Pulnix-9700 progressive scan cameras (or a Perceptron laser scanner as described in [7] mounted a pan-tilt mechanism, two image blob trackers (one for each camera) which is responsible for tracking target objects by controlling the pan-tilt mechanism, a submodule for computing depth from stereo, and a pose estimator. The vision system has been implemented on a combination of two Motorola 68040, and five i860 processors with physically distributed shared memory as main vehicles for communication among different submodules and with the state estimators. With the support of multi-processors, the vision system is capable of producing five estimated poses of a target object per second, which allow the state estimator to provide dynamic states of the target object at 100 Hz (using an extended Kalman filter) to the arm controller.

The integrated system is being extensively tested in a lab environment. The next planned mission for the EVAHR is to perform real-time vision-guided grasping of free flying objects in a reduced gravity environment (simulated by NASA's KC-135 aircraft) during the fourth quarter of 1995. The mission will be conducted in a modified KC-135 military aircraft flying a series of 40 to 50 parabolic trajectories with each trajectory giving a (approximately) 10 second period of micro-g environment.

4.7 Optical Computing for Space Exploration

Another line of research pursues the use of optical information processing for pattern recognition, landmark recognition, and terrain map construction. Researchers at NASA centers, specially Jet Propulsion Lab (JPL), Ames Research Center (ARC), and Johnson Space Center (JSC), are developing optical processors to perform specialized functions to be integrated with electronic numeric processors as part of efforts of the Mars Rover Sample Return (MRSR) program [5]. JPL has investigated the feasibility of using an optical correlator for landmark recognition [4]. In their work, an optical breadboard, consisting of a Hughes liquid crystal light valve, Fourier optics, and a thermoplastic holographic film was built and tested. A Vander Lugt matched spatial filter was synthesized in the thermoplastic film, which was used in successfully identifying a volcano inside a martian surface image obtained from an previous NASA Viking mission.

NASA Ames is developing new types of filters to be used in optical correlator systems [10], where both the input and filter planes are updated using programmable spatial light modulator (SLM) [16]. In the filter design, the modulation characteristics of the device onto which the filter is mapped are included in the synthesis equations. This system of nonlinear equations is then solved using an iteration procedure. Studies are also being conducted on the integration of optical processors into larger, heterogeneous computational systems. A database

of programmable filters arranged as a directed graph will be constructed. The output of the optical system will be used by a knowledge based system to determine the order in which to apply filters to rapidly traverse the database via a high speed filter controller.

A hybrid optical/digital system is under development at JSC for applications such as hazard identification and avoidance during the terminal phase of a planetary landing. Time-varying behavior in a sequence of images taken from a moving platform contains information on the 3D surface being approached. To extract that behavior, separating it from image jitter due to vehicle dynamics, an adaptive joint transform correlator has been employeed to compute (from two successive images) shift in correlation peak positions, which determine the offset address for the following digital image processor [22].

4.8 Vision System Planner and Vision Architecture

As mentioned earlier, space environment has been characterized as unknown and unpredictable. These characteristics call for a versatile and flexible machine vision system for intelligent robots to be able to function well in a dynamic environment. It is well known that no single type of sensors are suitable for all situations, and no single vision algorithm is robust enough to achieve desired performance for all cases. It is important for a vision system to be able to make decision on the choices of sensors and algorithms based on vision tasks to be carried out in different situations. A version of EVAHR's vision system planner has been designed and implemented at JSC [19]. Minimal requirements of the EVAHR's vision system planner includes

- on-line selection, adjustment or repositioning of sensors in order to acquire richer visual information,
- on-line selection of vision functions, vision algorithms, and environment/objects representation depending on missions, current situations, available knowledge about targets and environment.
- dynamic allocation of computing resources to speed up vision processing.

An example described in [19] demonstrates the EVAHR's vision system planner alternatively uses a video camera, a laser scanner, and the combination of the two for acquiring information about an unknown object, and activates various vision functions such as "searching", "tracking", "object identification", and pose estimation based on available knowledge about the observed object at each time instance. Detailed description of the EVAHR's vision system planner can be found in [19].

In order to meet requirements for processing multiple vision tasks simultaneously in real time, a real-time vision architecture is also under development at JSC. Due to the indeterministic nature of vision tasks, a hybrid computer architecture is utilized in the design of the EVAHR's vision system. The EVHAR's vision system is capable of performing various vision tasks in a mixture of pipelining, data partitioning and/or task partitioning modes at the same time using

all the available computing resources. Various processes may carry out tasks autonomously (by fetching tasks in a public task pool whenever they have completed previously fetched tasks), or perform assigned tasks through the coordination of a taskqueue server. Detailed discussion of the taskqueue server and parallel algorithms for many vision tasks can be found in [7].

4.9 Other Space Related Research

Other space related research activities on computer vision include operator vision aid for telerobotic assembly [2] [15], low power and low computation vision based navigation [13] (emphasizing reactive behavior), robotics in a controlled environment agriculture [21], and automatic Volcano localization [3], to name a few.

5 Concluding Remarks

This paper has given a list of target applications of computer vision in space robotics, discussed vision requirements for these applications, and described space related research on computer vision. Some critical issues need to be addressed before intelligent robots in general, and computer vision in particular, can be extensively applied in space environment. These issues include (but not limited) techniques in dealing with unfavorable illumination conditions (using artificial lights if necessary), real time vision processing in time critical applications (e.g. autonomous and precision landing, and vision-guided grasping of free floating objects), reliable, precise, and convenient geometric calibration between sensors, free flying space robots and constantly moving spacecraft platform, versatile vision architecture and vision task planners to handle more than one vision tasks using multiple sensors at the same time.

References

1. G. Anderson. Grasping a rigid object in zero-g. In *SPIE Conf. on Cooperative Intelligent Robotics in Space*, Boston, MA., Sept. 1993.
2. T. L. Brooks and Ilhan Ince. Operator vision aids for telerobotic assembly and servicing in space. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 886–891, Nice, France, May 1992.
3. M. C. Burl, U. M. Fayyad, P. Perona, P. Smyth, and M. P. Burl. Automating the hunt for volcanoes on venus. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 302–309, Seattle, WA., June 20-24 1994.
4. T. H. Chao and H. K. Liu. Real-time optical holographic tracking of multiple objects. *Applied Optics*, 28:226–, 1989.
5. T. H. Chao, E. Ochoa, and R. Juday. Optical information processing for nasa's space exploration. In *SPIE Proceedings Vol. 1151*, 1989.
6. Chiun-Hong Chien. An architecture for real-time vision processing. In *Proc. of Conf. on Intelligent Robotics in Field, Factory, Service, and Space*, pages 736–743, Houston, Tx., March 21-24 1994.

7. Chiun-Hong Chien. A computer vision system for extravehicular activity helper/retriever. *Applied Intelligence*, 5(3):251–268, July 1995.
8. E. G. Cooper, S. M. Jones, P. Goode, and S. L. Vazquez. Automated anomaly detection for orbiter high temperature reusable surface insulation. In *SPIE Conf. on Cooperative Intelligent Robotics in Space III*, pages 330–340, Boston, MA., November 1992.
9. E. D. Dickmanns. The 4-d approach to visual control of autonomous system. In *Proc. of Conf. on Intelligent Robotics in Field, Factory, Service, and Space*, pages 483–493, Houston, Tx., March 21-24 1994.
10. J. D. Downie. Optimal binary phase and amplitude correlation filters for polarization-rotating spatial light modulators. *Optical Engineering*, 32(8):1886–1894, August 1993.
11. J. D. Erickson and P. L. Graves. Space exploration initiative planet surface systems: Robotics systems design standards (rsds) (preliminary). Technical report, Automation and Robotics Division, NASA Johnson Space Center, Houston, TX 77058, 1991.
12. Jon Erickson et. al. An intelligent space robot for crew and equipment retrieval. *Applied Intelligence*, 5(1):7–40, July 1995.
13. A. S. Gavin and R. A. Brooks. Low computation vision-based navigation for a martian rover. In *Proc. of Conf. on Intelligent Robotics in Field, Factory, Service, and Space*, pages 685–695, Houston, Tx., March 21-24 1994.
14. L. Hewgill. Motion estimation of a freely rotating body in earch orbit. In *SPIE Conf. on Cooperative Intelligent Robotics in Space III*, Boston, MA., November 1992.
15. G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Rotex - the first remotely controlled robot in space. In *Proc. of IEEE Conf. on Robotics and Automation*, pages 2604–2611, San Diego, CA., May 8-13 1994.
16. D. A. Jaerd and d. J. Ennis. Inclusion of filter modulation in synthetic-discriminant-function construction. *Applied Optics*, 28:2343–, 1989.
17. In So Kewon. *Modeling Rugged Terrain by Mobile Robots with Multiple Sensors*. PhD thesis, Carnegie Mellon University, January 1991.
18. B. Krafft and H. Pien. Terrain reconstruction from rover images. Technical Report CSDL-P-3324, Draper Laboratory, Cambridge, MA., September 1993.
19. M. Magee, C. H. Chien, and T. W. Pendleton. A vision system planner for the extravehicular activity retriever. In *Intern. Conf. on Intelligent Autonomous System III*, Pittsburgh, PA., February 1993.
20. M. Magee, W. Hoff, L. Gatrell, C. Sklair, and W. Wolfe. Employing sensor repositioning to refine spatial reasoning in an industrial robotic environment. *Trans. of the ASAE*, 36(2):593–603, 1993.
21. G. E. Miles and J. D. Erickson. Robotics in a controlled environment agriculture. In *Proc. of Conf. on Intelligent Robotics in Field, Factory, Service, and Space*, pages 598–605, Houston, Tx., March 21-24 1994.
22. E. C. Tam, D. A. Gregory, and R. D. Juday. Autonomous real-time object tracking with an adaptive joint transform correlator. *Optical Engineering*, 29(4):314–320, April 1990.
23. R. Volpe and J. Balaram. Technology for robotics surface inspection in space. In *Proc. of Conf. on Intelligent Robotics in Field, Factory, Service, and Space*, pages 131–141, Houston, Tx., March 21-24 1994.

Motion Estimation and Segmentation

Optical-Flow Estimation while Preserving its Discontinuities: A Variational Approach

Rachid Deriche[1] - Pierre Kornprobst[1] - Gilles Aubert[2]

[1] INRIA, 2004 route des Lucioles, BP 93
F-06902 Sophia-Antipolis Cedex, France
Email: der@sophia.inria.fr
[2] Laboratoire J.A Dieudonné - URA 168 du CNRS
Universitè de Nice-Sophia Antipolis
06108 Nice-Cedex 2 (France)

Abstract. This paper describes a variational approach devised for the purpose of estimating optical flow from a sequence of images with the constraint to preserve the flow discontinuities. This problem is set as a regularization and minimization of a non quadratic functional. The *Tikhonov* quadratic regularization term usually used to recover smooth solution is replaced by a particular function of the gradient flow specifically derived to allow flow discontinuities formation in the solution. Conditions to be fulfilled by this specific regularizing term, to preserve discontinuities and insure stability of the regularization problem, are also derived. To minimize this non quadratic functional, two different methods have been investigated. The first one is an iterative scheme to solve the associated non-linear Euler-Lagrange equations. The second solution introduces dual variables so that the minimization problem becomes a quadratic or a convex functional minimization problem. Promising experimental results on synthetic and real image sequences will illustrate the capabilities of this approach.

1 Introduction

In this last decade, numerous methods have been proposed to compute 2D optical flow [8, 9, 11, 7, 4, 3, 12, 13] . Almost all these approaches use the classical constraint equation that relates the gradient of brightness to the components u and v of the local flow to estimate the optical flow. Because this problem is ill-posed, additional constraints are required. The most used one is to add a quadratic smoothness constraint as done originally by Horn and Schunk [8]. However, in order to estimate the optical flow more accurately, other constraints involving high order spatial derivatives have also been used [12]. Nevertheless, several of the proposed methods lacked robustness to the presence of occlusion, and yielded smooth optical flow. The variational approach proposed in this paper is motivated by the need to recover the optical flow while preventing the method from trying to smooth the solution across the flow discontinuities. To cope with discontinuities, several methods have been proposed [4, 13, 7]. The method presented here is inspired from a recent framework that has proven to

be very useful in some image processing tasks as image restoration [14, 16, 2, 5]. It is a variational approach devised for the purpose of estimating optical flow from a sequence of images with the constraint to preserve the flow discontinuities. This problem is set as a regularization and minimization of a non quadratic functional. The *Tikhonov* quadratic regularization term usually used to recover smooth solution is replaced by a particular function of the gradient flow specifically derived to allow flow discontinuities formation in the solution.

2 Statement of the Problem

Let $I(x,y,t)$ be the grey level intensity at time t at the image point (x,y) and $u(x,y)$ and $v(x,y)$ denote the x and y components of the optical flow vector $\mathbf{v}$ at that point. Assuming that the grey level intensities are the same at time t and $t+\delta t$ leads to the well known *Optical Flow Constraint* :

$$\nabla I(x,y,t)\cdot\mathbf{v}+I_t(x,y,t)=0 \tag{1}$$

Where ∇ is the gradient operator, and I_t denotes the temporal derivative of $I(x,y,t)$. The scalar equation (1) only permits the calculation of the components of the velocity field which is parallel to the spatial gradient ∇I (*aperture problem*). Additional constraints are therefore required to reduce the space of admissible functions. The most popular constraint is the *smoothness constraint* originally introduced by *Horn* and *Schunck* [8]. This approach combines the *Optical Flow Constraint* with the quadratic smoothing term of *Tikhonov* [15] and considers the optical flow as the global minimum of the functional:

$$F(\mathbf{v})=\int_{\Omega}(\alpha C(\mathbf{v})+S(\mathbf{v}))d\Omega \tag{2}$$

where α is a positive coefficient that weights the error in the image motion equation relative to the departure from smoothness. Ω denotes the image domain and the 2 terms of eq. (2) are given by :

$$\begin{cases} C(\mathbf{v})=(I(x,y,t)\cdot\mathbf{v}+I_t(x,y,t))^2 \\ S(\mathbf{v})=\mid\nabla u\mid^2+\mid\nabla v\mid^2 \end{cases}$$

The minimization of this functional can be performed by writing the associated Euler-Lagrange equations :

$$\begin{cases} \nabla^2 u=\alpha I_x(I_x u+I_y v+I_t) \\ \nabla^2 v=\alpha I_y(I_x u+I_y v+I_t) \end{cases} \tag{3}$$

where

$$\nabla^2=\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}$$

is the Laplacian operator. Boundary conditions are then added and a finite difference scheme is usually used in the discrete case to solve iteratively this coupled pair of elliptic second-order partial differential equation. Unfortunately,

the use of a quadratic term for $S(\mathbf{v})$ leads to a continuous and smooth solution for the optical flow i.e a solution where discontinuities are not preserved. This is not really desirable if one wants to preserve the flow discontinuities and recover the optical flow as accurately as possible because such discontinuities can be present for instance on silhouettes, where one object occludes another or between two different moving objects. The next section is devoted to presenting our approach to tackle in an efficient way this important problem.

3 A Variational Approach to Preserve Discontinuities

In order to preserve the flow discontinuities while regularizing the solution, several methods have been proposed. Some of these approaches allow to capture these discontinuities by judiciously choosing the α parameter of equation 2. Other approaches take into account gradient information of the optical flow field and recently interesting variational approaches have also been proposed to deal with this class of problem. The reader interested can refer to [4, 13, 7] where original solutions are proposed to tackle this important problem.

In the following subsection, we summarize our variational approach to optical flow estimation. This approach is inspired from the approaches developed for image restoration purpose in [14, 16]. Very recently, existence and uniqueness results have been demonstrated in [2], also for the image restoration problem. A detailed review of all these approaches can be found in [5].

3.1 The Energy Function

To cope with discontinuities, a natural way to proceed is to forbid regularizing and smoothing across such flow discontinuities. One way of taking into account these technical remarks is to consider a function $\phi(.)$ that preserves the discontinuities :

$$S(\mathbf{v}) = \phi(\mid \nabla u \mid) + \phi(\mid \nabla v \mid) \tag{4}$$

For example, a quadratic function as proposed by *Tikhonov* in [15], clearly does not preserve the discontinuities and must therefore be avoided.

Our key idea to deal with this problem is to write the Euler-Lagrange equations associated to this new functional. This leads to the following coupled pair of second order partial differential equations :

$$\begin{cases} div(\frac{\phi'(|\nabla u|)}{|\nabla u|}\nabla u) = 2\alpha(I_x u + I_y v + I_t)I_x & \\ & (x,y)\ \epsilon\Omega \\ div(\frac{\phi'(|\nabla v|)}{|\nabla v|}\nabla v) = 2\alpha(I_x u + I_y v + I_t)I_y & \\ \frac{\phi'(|\nabla u|)}{|\nabla u|}\frac{\partial u}{\partial n} = 0 \ , \ \frac{\phi'(|\nabla v|)}{|\nabla v|}\frac{\partial v}{\partial n} = 0 & (x,y)\epsilon\ \partial\Omega \end{cases} \tag{5}$$

where div denotes the classical divergence operator, ϕ' represents the first derivative of $\phi(s)$ with respect to the parameter s and n indicates the unit

vector normal to the boundary $\partial\Omega$. Notice that the main difference with the equations 3 involving the *Tikhonov* regularization term is that the Laplacian term has been changed by the following divergence term :

$$2\nabla^2 w \longrightarrow div\left(\frac{\phi'(\mid \nabla w \mid)}{\mid \nabla w \mid}\nabla w\right) \tag{6}$$

for w being u or v. It is this replacement of the Laplacian operator by a divergence operator that will allow us to develop an anisotropic smoothing process of the flow field.

Developing and simplifying the divergence term, it can be shown that it is equal to the sum of two terms:

$$div(\frac{\phi'(\mid \nabla w \mid)}{\mid \nabla w \mid}\nabla w) = \underbrace{\frac{\phi'(\mid \nabla w \mid)}{\mid \nabla w \mid}}_{c_\xi} w_{\xi\xi} + \underbrace{\phi''(\mid \nabla w \mid)}_{c_\eta} w_{\eta\eta} \tag{7}$$

where $\phi'(s)$ and $\phi''(s)$ indicate respectively the first and second derivatives of $\phi(s)$ with respect to s, $w_{\eta\eta}$ is the second order directional derivative of w in the direction $\eta = \frac{\nabla w}{|\nabla w|}$ and $w_{\xi\xi}$ is the second order directional derivative in the direction ξ orthogonal to the gradient.

In order to regularize the solution and preserve optical flow discontinuities, one would like to smooth isotropically the optical flow field inside homogeneous regions and preserve the flow discontinuities in the inhomogeneous regions. Assuming that the function $\phi''(.)$ exists,the condition on smoothing in an isotropic way inside homogeneous regions can be achieved by imposing the following conditions on the $\phi(.)$ function:

$$\lim_{|\nabla_m w|\to 0} \frac{\phi'(\mid \nabla_m w \mid)}{\mid \nabla_m w \mid} = \lim_{|\nabla_m w|\to 0} \phi''(\mid \nabla_m w \mid) = \phi''(0) > 0 \tag{8}$$

Therefore, at the points where $\mid \nabla w \mid$ is small, the divergence term becomes

$$div\left(\frac{\phi'(\mid \nabla w \mid)}{\mid \nabla w \mid}\nabla w\right) = \phi''(0)(w_{\xi\xi} + w_{\eta\eta}) = \phi''(0)\nabla^2 w. \tag{9}$$

This case corresponds to the situation where the function $\phi(.)$ is quadratic and the smoothing is isotropic. Note that the coefficient $\phi''(0)$ is required to be positif, otherwise the regularization part will act as an inverse heat equation notably known as an instable process. Therefore, the following coupled pair of second order partial differential equations will have to be solved :

$$\begin{cases} \phi''(0)\nabla^2 u = 2\alpha(I_x u + I_y v + I_t)I_x & \\ & (x,y)\ \epsilon\Omega \\ \phi''(0)\nabla^2 v = 2\alpha(I_x u + I_y v + I_t)I_y & \\ \frac{\partial u}{\partial n} = 0 \ , \ \frac{\partial v}{\partial n} = 0 & (x,y)\epsilon\ \partial\Omega \end{cases} \tag{10}$$

In order to preserve the flow discontinuities near inhomogeneous regions presenting a strong flow gradient, one would like to smooth along the isophote (curve

with constant flow) and not across them. This leads to stopping the diffusion in the gradient direction η, i.e. setting the weight $\phi''(\mid \nabla w \mid)$ to 0, while keeping a stable diffusion along the direction orthogonal ξ, i.e. setting the weight $\frac{\phi'(|\nabla w|)}{|\nabla w|}$ to some positive constant:

$$\lim_{|\nabla w| \to \infty} \phi''(\mid \nabla_m w \mid) = 0 \lim_{|\nabla w| \to \infty} (\frac{\phi'(\mid \nabla_m w \mid)}{\mid \nabla_m w \mid}) = \beta > 0 \tag{11}$$

Therefore, at the points where the flow gradient is strong, u and v will be the solutions of the following equations :

$$\begin{cases} \beta u_{\xi\xi} = 2\alpha(I_x u + I_y v + I_t) I_x & \\ & (x,y) \in \Omega \\ \beta v_{\xi\xi} = 2\alpha(I_x u + I_y v + I_t) I_y & \\ \frac{\partial u}{\partial n} = 0 \ , \ \frac{\partial v}{\partial n} = 0 & (x,y) \in \partial\Omega \end{cases} \tag{12}$$

Note that the positiveness of the β coefficient is also required to generate a stable smoothing process in the ξ direction.

Unfortunately, the two conditions of (11) cannot be satisfied simultaneously by a function $\phi(\mid \nabla_m Z \mid)$. However, the following conditions can be imposed in order to decrease the effects of the diffusion along the gradient more rapidly than those associated with the diffusion along the isophotes:

$$\begin{array}{l} \lim_{|\nabla_m w| \to \infty} \phi''(\mid \nabla_m w \mid) = \lim_{|\nabla_m w| \to \infty} \frac{\phi'(|\nabla_m w|)}{|\nabla_m w|} = 0 \\ \lim_{|\nabla_m w| \to \infty} \frac{\phi''(|\nabla_m w|)}{\frac{\phi'(|\nabla_m w|)}{|\nabla_m w|}} = 0 \end{array} \tag{13}$$

The conditions given by Equations (8) and (13) are those which one has to impose in order to deal with a regularization process which preserves the discontinuities. As it has been shown very recently in [2], these conditions are also sufficient to prove that the model is well posed mathematically, and the existence and uniqueness of a solution is also guaranteed by these conditions.

One can easely see that only the last two functions fulfill all the conditions mentioned above. Moreover, the convex potential functions allows to obtain uniqueness of a global minimum.

3.2 Energy Minimization

The energy function given by eq. (2) is minimized by solving the coupled pair of partial differential equations given by eq. (5). In his subsection, we present 2 different numerical schemes that have been developed for moving iteratively towards the solution.

Author	$\phi(s)$	$\frac{\phi'(s)}{s}$
Geman et Reynolds	$\frac{s^2}{1+s^2}$	$\frac{2}{(1+s^2)^2}$
Malik et Perona	$\log(1+s^2)$	$\frac{2}{(1+s^2)}$
Green	$2\log[\cosh(s)]$	$\begin{cases} 2 & s=0 \\ 2\tanh(s)/s & s\neq 0 \end{cases}$
Aubert	$2\sqrt{1+s^2}-2$	$\frac{2}{\sqrt{(1+s^2)}}$

An Implicit Scheme The first scheme that has been developed to solve the Euler-Lagrange equations of eq. (5) uses a linearized version of the discrete equations and a Gauss-Seidel relaxation method for moving iteratively towards the solution of this problem.

Implicit Scheme:

It can be shown that the divergence operator can be written as follows [10]

$$\left(div(\frac{\phi'(\mid \nabla w \mid)}{\mid \nabla w \mid}\nabla w)\right)_{i,j} \simeq \sum_{p,q\epsilon\{-1,0,1\}^2} \sigma_{wi+p,j+q} w_{i+p,j+q} \tag{14}$$

where the coefficients $\sigma_{wi+p,j+q}$ depend on the values of the 8 neighbors.

Using this result, the coupled pair of PDE of eq. (5) may be rewritten as the following non-linear system of equations

$$\begin{cases} \sum \sigma_{ui+p,j+q} u_{i+p,j+q} = 2\alpha\left((I_x u + I_y v + I_t)I_x\right)_{i,j} \\ \sum \sigma_{vi+p,j+q} v_{i+p,j+q} = 2\alpha\left((I_x u + I_y v + I_t)I_y\right)_{i,j} \\ \oplus \quad \text{Boundary conditions on the flow} \end{cases} \tag{15}$$

This large system of equations in non linear due to the fact that the divergence term includes the function to be estimated (i.e u and v). There are numerous iterative schemes for the solution of a large sparse set of equations, among them Gauss Seidel and Jacobi are the most popular. We use a linearized version of the discrete equations and we apply a Gauss-Seidel relaxation method for moving iteratively towards the solution of this problem.

Half quadratic regularization In the previous subsection, a non quadratic energy functional was minimized by solving non linear equations. In this subsection, instead of stress upon solving a non linear system, we introduce a new function which, although defined over an extended domain, has the same minimum in $\mathbf{v}$ as F and can be manipulated with linear algebraic methods.

We first start by presenting an important theorem inspired from *Geman* and *Reynolds* theorem (see [6]). A proof of this theorem can be found in [1].

Theorem 3.2.1
Let $\phi : [0,+\infty[\rightarrow [0,+\infty[$ be such that $\phi(\sqrt{x})$ is strictly concave on $]0,+\infty[$. Let L and M be defined as:

$$L = \lim_{x\rightarrow+\infty} \frac{\phi'(x)}{2x} \quad and \quad M = \lim_{x\rightarrow 0^+} \frac{\phi'(x)}{2x}$$

Then, we have the following properties:

- *there exists a strictly convex and decreasing function $\psi : [L,M] \rightarrow [\alpha,\beta]$ where*

$$\phi(x) = \lim_{L\leq b\leq M} \inf (bx^2 + \psi(b)) \tag{16}$$

  such that:

$$\alpha = \lim_{x\rightarrow+\infty} \left(\phi(x) - x^2\frac{\phi'(x)}{2x}\right) \; et \; \beta = \lim_{x\rightarrow 0_+} \phi(x)$$

- *for every fixed $x \geq 0$ the value b_x for which the minimum is reached is unique and given by:*

$$b_x = \frac{\phi'(x)}{2x} \tag{17}$$

The variable b_x is called an auxiliary variable associated to x. One can check that this theorem can be applied to all the $\phi(.)$ functions proposed for discontinuities preservations. So, introducing two dual variables b_u and b_v associated respectively to $\mid \nabla u \mid$ and $\mid \nabla v \mid$ and noting $\mathbf{b} = \begin{pmatrix} b_u \\ b_v \end{pmatrix}$, the problem becomes to minimize the resulting augmented energy with respect to $\mathbf{b}$ and $\mathbf{v}$:

$$F^*(\mathbf{v},\mathbf{b}) = \int_\Omega C(\mathbf{v}) + \alpha S^*(\mathbf{v},\mathbf{b})\, d\Omega$$

where

$$S^*(\mathbf{v},\mathbf{b}) = b_u \mid \nabla u \mid^2 + \psi(b_u) + b_v \mid \nabla v \mid^2 + \psi(b_v)$$

The advantage of considering F^* instead of F is twofold:
- for a fixed V, the functional F^* is convex in $\mathbf{b}$
- for a fixed $\mathbf{b}$, the functional F^* is convex in V

These properties are used to perform the algorithm described in table 1.
The two stages of minimization work as follows :
- calculating $\mathbf{b}^{n+1} = arg \min_{\mathbf{b}} F^*(V^n,\mathbf{b})$ amounts solving:

$$\min_{\mathbf{b}=\begin{pmatrix} b_u \\ b_v \end{pmatrix}} \int_\Omega b_u \mid \nabla u \mid^2 + \psi(b_u) + b_v \mid \nabla v \mid^2 + \psi(b_v)\, d\Omega$$

$V^0 \equiv \mathbf{0}$	
Repeat	
	$\mathbf{b}^{n+1} = arg\min_{\mathbf{b}} F^*(V^n, \mathbf{b})$
	$V^{n+1} = arg\min_V F^*(\mathbf{v}, \mathbf{b}^{n+1})$
	$n = n + 1$
Until	
convergence	

Table 1. Half quadratic regularization principle

The two variables b_u and b_v being independent, we can use theorem 3.2.1 and show that the minimum is reached for:

$$b_u = \frac{\phi'(\mid \nabla u \mid)}{2 \mid \nabla u \mid} \quad , b_v = \frac{\phi'(\mid \nabla v \mid)}{2 \mid \nabla v \mid} \tag{18}$$

- calculating $\mathbf{v}^{n+1} = arg\min_V F^*(\mathbf{v}, \mathbf{b}^{n+1})$ is equivalent solving:

$$\min_{u,v} \int_\Omega \alpha(\nabla I.\mathbf{v} + I_t)^2 + (b_u^F \mid \nabla u \mid^2 + b_\mathbf{v}^F \mid \nabla v \mid^2)\, d\Omega$$

where we noted b_w^F to recall that the auxiliary variables are now fixed. This is now easier because the functional to be minimized is quadratic and hence we only have to solve linear equations. This can be done iteratively by the classical Gauss-Seidel method. The associated Euler-Lagrange equations can be written as follows :

$$\begin{cases} div(b_u^F \nabla u) = \alpha(I_x u + I_y v + I_t) I_x \\ div(b_v^F \nabla v) = \alpha(I_x u + I_y v + I_t) I_y \\ \oplus \text{ Boundary conditions on the flow} \end{cases}$$

This algorithm converges to the unique solution of our problem. The reader interested by more details may refer to [2] where convergence results have been proved.

4 Experimental results

To validate the approach that has been presented in this paper, some experiments have been carried out using synthetic and real image sequences extracted from [3].

For each experiment (see next page), the left figure is the image for which the flow is computed, the middle one is the result obtained with a quadratic regularization term and the right figure corresponds to the result computed

with the potential function proposed by *Aubert*. Parameters values are written in the caption. As expected, it appears from the figures that the discontinuities are better preserved using the specific *Aubert* function.

References

1. G. Aubert, M. Barlaud, L. Blanc-Feraud, and P. Charbonnier. Deterministic edge-preserving regularization in computed imaging. *IP2.3 Computed imaging*, 1994.
2. G. Aubert and L. Lazaroaia. A variational method in image recovery. TR 423, URA au CNRS no 168,laboratoire J-A Dieudonne, Université de Nice Sophia-Antipolis-CNRS, June 1995.
3. J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *The International Journal of Computer Vision*, 12(1):43–77, 1994.
4. J. Cohen. Nonlinear variational method for optical flow computation. In *Proceedings 8th SCIA*, volume 1, pages 523–530, 1993.
5. R. Deriche and O. Faugeras. Les EDP en traitement des images et vision par ordinateur. Technical report, INRIA, 1995.
6. D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):367–383, 1993.
7. F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, December 1993.
8. Berthold K. P. Horn and Brian G. Schunk. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.
9. Berthold Klaus Paul Horn. *Robot Vision*. MIT Press, 1986.
10. P Kornprobst. Calcul de Flot Optique avec Préservation des Discontinuités. Rapport de stage de DEA. Proposé et encadré par R. Deriche et G. Aubert, INRIA, Université Nice-Sophia Antipolis, France, Juin 1995.
11. Hans-Hellmut Nagel. Direct estimation of optical flow and of its derivatives. In G.A. Orban and H.-H. Nagel, editors, *Artificial and Biological Vision Systems*, Basic Research Series, pages 191–224. Springer-Verlag, 1992.
12. Michael Otte and Hans-Helmut Nagel. Optical flow estimation: Advances and comparisons. In Jan-Olof Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume 800 of *Lecture Notes in Computer Science*, pages 51–70. Springer-Verlag, 1994.
13. M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of Optical Flow and its Discontinuities using Non-Linear Diffusion. In *Proceedings of the 3rd ECCV, II*, number 801 in Lecture Notes in Computer Science, pages 295–304. Springer Verlag, 1994.
14. L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
15. A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems*. Winston and Sons, Washington, D.C., 1977.
16. Y.L. You, M. Kaveh, W.Y. Xu, and A. Tannenbaum. Analysis and Design of Anisotropic Diffusion for Image Processing. In *Proc. IEEE Int. Conf. on Image Proc. ICIP-94*, volume II of III, pages 497–501, Austin(Texas), November 13-16 1994.

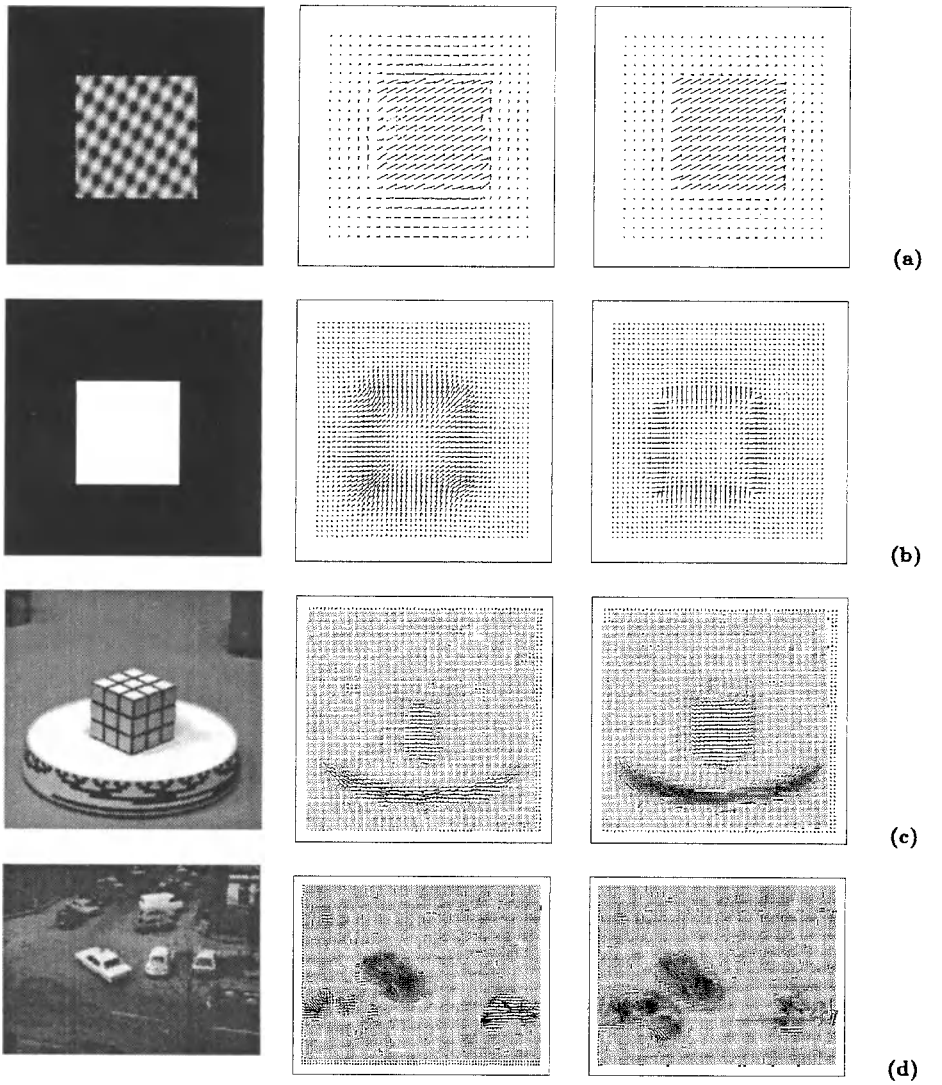


Fig. 1. (a) plaid pattern moving in a window: 100 iterations, $\alpha_H = 200$, $\delta = 0.5$ (b) translating square: 100 iterations, $\alpha_H = 10000$, $\delta = 1.0$ **(c)** Rubik's cube: 100 iterations. $\alpha_H = 10000$, $\delta = 0.5$ **(d)** Taxi data: 100 iterations, $\alpha_H = 10000$, $\delta = 0.7$

Space- and Time-Variant Estimation Approaches and the Segmentation of the Resulting Optical Flow Fields

H.-H. Nagel [†] [‡], *A. Gehrke* [†], *M. Haag* [†], *and M. Otte* [†]

[†] Institut für Algorithmen und Kognitive Systeme
Fakultät für Informatik der Universität Karlsruhe (TH)
Postfach 6980, D-76128 Karlsruhe / Germany
and
[‡] Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB)
Fraunhoferstr. 1, D-76131 Karlsruhe / Germany (e-mail: hhn@iitb.fhg.de)

Abstract. A 'Gaussian-blob model' for local spatiotemporal gray value variations is exploited in order to simultaneously estimate an optical flow vector and the dominant spatiotemporal scale of the gray value variation which forms the basis for this estimation. A two-step, non-iterative, local estimation approach is developed which does not include any explicit smoothness term. Various experiments with real world image sequences nevertheless yield smooth optical flow fields with surprisingly good results, even close to discontinuities of the optical flow field along occluding contours. 'Close' in this context means within one to three pixels, i. e. much less than the width of the weight masks employed to estimate the spatiotemporal gradient of the gray value variation.

1 Introduction

Optic flow – the apparent shift velocity $(u_1, u_2)^T$ of a gray value structure in the image plane – is usually *defined* by the requirement that the gray value $g(x, y, t)$ is locally stationary as a function of the image plane coordinates $(x, y)^T$ and time t: $dg(x, y, t) = 0$. This results in the so-called 'Optic Flow Constraint Equation (OFCE)' [3]

$$\frac{\partial g(x,y,t)}{\partial x} u_1 + \frac{\partial g(x,y,t)}{\partial y} u_2 + \frac{\partial g(x,y,t)}{\partial t} 1 = 0 \quad or$$
$$(\nabla g)^T \mathbf{u} = 0 \tag{1}$$

with $\mathbf{u} = (u_1, u_2, 1)^T$. Notions related to optical flow such as the aperture problem, differential versus feature based estimation approaches, the difference between optic flow and image motion, etc. are explained in course books, for example [3]. Current estimation approaches have been well reviewed by Barron et al. 94 [1]. These latter authors employed synthetic image sequences or semi-synthetic ones, generated from a single digitized image frame by image manipulations, in order to compare the results obtained by various approaches with ground truth. In distinction to [1], [10] applied various local differential optic flow estimation

approaches to a real world image sequence of a *genuinely three-dimensional polyhedral scene* and compared their results with ground truth about image motion. As long as the object surfaces are well textured and the scene structure does not cause discontinuities of image motion, they find that the optic flow fields estimated by a number of local differential approaches agree with the image motion ground truth within 0.1 pixel per frame. Systematic discrepancies are observed at discontinuities of the image motion field.

These results stimulated us to investigate three fundamental questions related to current differential estimation approaches:

- The assumption of a locally stationary gray value structure oversimplifies most spatiotemporal gray value variations even if it sometimes allows to estimate surprisingly good approximations to image motion. Is it possible to introduce a definition of optic flow which does not rely on this assumption?
- What exactly means 'local', i. e. what is the optimal scale for the estimation of optic flow at each image location ?
- Provided we introduce a concept of 'locality', can we exploit it in order to algorithmically estimate the necessary parameters from the image sequence itself rather than having to tune the estimation process 'by hand' ?

[8] extended a space-variant texture estimation approach based on the evaluation of an averaged spatial gradient outer product tensor field ([12]). Starting from [8], we report about experimental investigations with real world image sequences regarding various aspects of an approach which differs from other, similar ones ([4, 11, 7]) by the explicit introduction of a *model* for the local spatiotemporal gray value variation $\mathrm{g}(\mathbf{x})$ with $\mathbf{x} = (x, y, t)^T$. This enables us to study the interaction between the actual $\mathrm{g}(\mathbf{x})$ and the spatiotemporal scale of the filters employed to estimate $\nabla \mathrm{g}$, in particular in the vicinity of optical flow field discontinuities.

2 A Picture Domain Model of Local Gray Value Variation

The idea underlying the approach of [8] is formulated as an explicit picture domain model of the local spatiotemporal gray value variation. Stimulated by the work of [5], this model is represented as a '3D Gaussian blob' defined by $g(\mathbf{x}) = g(0)(2\pi)^{-3/2}|\Sigma_0|^{-1/2}\, e^{-\frac{1}{2}\mathbf{x}^T \Sigma_0^{-1}\mathbf{x}}$. Size, shape, and orientation of this blob are specified by the symmetric, positive semidefinite matrix Σ_0. Let $\mathbf{e}_i$ denote the i-th eigenvector of Σ_0 and σ_i^2 the corresponding eigenvalue, with $\sigma_1^2 \leq \sigma_2^2 \leq \sigma_3^2$. Σ_0 can then be written in the form

$$\Sigma_0 = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} \begin{pmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{pmatrix} \quad . \tag{2}$$

Let us assume for the moment that eigenvector $\mathbf{e}_3 = (e_{31}, e_{32}, e_{33})^T$ is aligned with the t-axis. Provided $\sigma_3 = \infty$, such a Σ_0 corresponds to a temporally stationary gray value blob. The curves of constant gray value are ellipses with minor

half-axis proportional to σ_1, oriented in the direction of eigenvector $\mathbf{e}_1$. The major half-axis is proportional to σ_2 and oriented in the direction of eigenvector $\mathbf{e}_2$. Large values for σ_1 or σ_2 correspond to small gray value variations in the direction of the eigenvectors $\mathbf{e}_1$ or $\mathbf{e}_2$, respectively.

If, however, the third eigenvector $\mathbf{e}_3$ deviates from the t-axis, the center of the elliptic curves of constant gray value at each time t will be shifted along the direction given by the projection of $\mathbf{e}_3$ into the (x,y)-image plane. Such a curve of constant gray value may be considered as a gray value structure shifting in the image plane with an apparent velocity proportional to the vector

$$\mathbf{u} = \left(\frac{e_{31}}{e_{33}}, \frac{e_{32}}{e_{33}}, \frac{e_{33}}{e_{33}} = 1 \right)^T . \tag{3}$$

Admission of finite values for σ_3 models a Gaussian variation of the intensity as a function of time, i. e. we *no longer restrict* ourselves to the assumption of temporally *constant* gray values. Σ_0 thus defines the local spatiotemporal scale and orientation of a gray value structure as well as its shift velocity in the image plane.

3 Estimation Approach for Scale and Optic Flow

The model for the gray value variation introduced in the preceding Section will now be exploited in order to derive a procedure for the estimation of the local spatiotemporal scale and of the optical flow vector. The mathematical background for this procedure will be outlined in the following three steps:

1. Determine a low-pass filtered version of the gray value structure and of its gradient. Convolving a 3D Gaussian blob with a low-pass filter given by a trivariate Gaussian $(2\pi)^{-\frac{3}{2}}|\Sigma|^{-\frac{1}{2}}e^{-\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}}$ yields

$$g_{low-pass}(\mathbf{x}) = \frac{g(0)\ \ e^{-\frac{1}{2}\mathbf{x}^T(\Sigma_0+\Sigma)^{-1}\mathbf{x}}}{(2\pi)^{3/2}\sqrt{|\Sigma_0+\Sigma|}} \quad . \tag{4}$$

   The gray value gradient of $g_{low-pass}(\mathbf{x})$ is then given by

$$\nabla g_{low-pass}(\mathbf{x}) = -(\Sigma_0+\Sigma)^{-1}\mathbf{x}\ g_{low-pass}(\mathbf{x}) \quad . \tag{5}$$

2. Evaluate an integral of the outer product of the gradient vector with itself, weighted by a Gaussian, and inversely weighted by the low-pass filtered version of the gray value function.
3. Solve the resulting weighted 'Gradient vector Outer Product Tensor (GOPT)' for $\Sigma_0(\mathbf{x})$.

We define the weighted 'Gradient vector Outer Product Tensor (GOPT)' as

$$GOPT = \iiint_{-\infty}^{+\infty} d\xi \left(\frac{\nabla g_{low-pass}(\xi)\ \ (\nabla g_{low-pass}(\xi))^T}{g_{low-pass}(\xi)^2} \frac{1}{(2\pi)^{3/2}\sqrt{|\Sigma|}}\ e^{-\frac{1}{2}\xi^T\Sigma^{-1}\xi} \right). \tag{6}$$

Insertion of the expression for $\nabla g_{low-pass}(\mathbf{x})$ results in

$$GOPT = (\Sigma_0 + \Sigma)^{-1} \; \Sigma \; \left((\Sigma_0 + \Sigma)^{-1}\right)^T \quad . \tag{7}$$

We note that Σ as well as $\Sigma_0 + \Sigma$ are symmetric. We may thus omit the transpose operation for the last factor on the right hand side of equ. (7). Using $\Sigma^{\frac{1}{2}} = diag(\sigma_x, \sigma_y, \sigma_t)$, we may write

$$\begin{aligned} &\Sigma^{\frac{1}{2}} \; GOPT \; \Sigma^{\frac{1}{2}} \\ &= \Sigma^{\frac{1}{2}} \; (\Sigma_0 + \Sigma)^{-1} \; \Sigma^{\frac{1}{2}} \; \Sigma^{\frac{1}{2}} \; (\Sigma_0 + \Sigma)^{-1} \; \Sigma^{\frac{1}{2}} \; . \end{aligned} \tag{8}$$

Since Σ and thus $\Sigma^{\frac{1}{2}}$ is known, we obtain the estimate $\widehat{\Sigma_0}$ for the matrix which characterizes the local spatiotemporal gray value variation:

$$\widehat{\Sigma_0} = \Sigma^{\frac{1}{2}} \left(\Sigma^{\frac{1}{2}} \; GOPT \; \Sigma^{\frac{1}{2}} \right)^{-\frac{1}{2}} \; \Sigma^{\frac{1}{2}} - \Sigma \quad . \tag{9}$$

Provided our model of the local gray value variation as a spatiotemporal Gaussian blob is a good approximation of the observed data, we thus obtain an estimate of the local gray value variation in a manner which is (approximately) independent of the Gaussian filter Σ employed to estimate the gray value gradient.

In general, $\widehat{\Sigma_0}$ will not be diagonal: only in rare cases will the local gray value structure be aligned with the image coordinate system (x,y,t). It is straightforward, however, to transform $\widehat{\Sigma_0}$ into a representation analogous to equ. (2). This simultaneously yields an estimate of the optic flow vector $\mathbf{u}$ according to equ. (3) if we pick the eigenvector associated with the largest eigenvalue. This latter requirement corresponds to our definition of optic flow as indicating the shift velocity associated with *minimal* – in distinction to the, in general unrealistic, requirement of vanishing – changes of the local gray value structure as a function of time.

We thereby have obtained answers to the fundamental questions posed in the introduction, yielding a closed form solution for a nonlinear problem. We can, moreover, study how the solution is modified if we consider a deviation of the actual gray value structure from the Gaussian blob model. Inspection of equ. (6) immediately shows that the matrix inversion implied by the transition from equ. (8) to equ. (9) will be endangered by an ill-conditioned GOPT in the case where $\nabla g_{low-pass}(\mathbf{x})$ becomes small. In order to maintain stability even for nearly homogeneous gray value areas, we add a constant multiple $\epsilon^2 I$ of the unity matrix I to the numerator of the integrand in equ. (6). The result obtained by this modification will be denoted by $\widehat{\Sigma_0'}$.

In case of slow gray value variations in one or more dimensions, the eigenvalues of $\widehat{\Sigma_0'}$ may enforce an extent of Gaussian filter masks which could become impractical. We therefore reset the maximum eigenvalue σ_3^2 of $\widehat{\Sigma_0'}$ to 4 if it exceeds this limit. The two remaining eigenvalues σ_1^2 and σ_2^2 are rescaled to $max\{1, 4\frac{\sigma_1^2}{\sigma_3^2}\}$ and $max\{1, 4\frac{\sigma_2^2}{\sigma_3^2}\}$, respectively. The lower limit of 1 is enforced in

order to obtain a suitable digital approximation of a Gaussian and of its partial derivatives.

Since $\widehat{\Sigma_0'}$ is taken to describe our estimate of the local spatiotemporal gray value variation, we substitute it for the space-invariant Σ chosen a priori and repeat the procedure outlined above, with the resulting estimate denoted by $\widehat{\Sigma_0''}$.

It has been observed that the modified approach comprising a term $\epsilon^2 I$ in the definition of GOPT exhibits a weak bias towards an optical flow estimate with zero magnitude in gray value areas which are approximately homogeneous along one or two dimensions, provided the eigenvalue σ_t^2 corresponding to the temporal component of Σ is chosen somewhat larger than the spatial eigenvalues σ_x^2 and σ_y^2 ([2]). A symbolic sensitivity analysis (see [8]) allows it to understand this effect quantitatively.

4 Results and Conclusions

We have applied the approach outlined in the preceding Section to digitized real world interlaced image sequences obtained by a stationary video camera recording road traffic – see, e.g., Figure 1.

Fig. 1. Image frame No. 24 from the image sequence 'Ettlinger-Tor-Platz' in Karlsruhe / Germany, from [6]. Due to space limitations, we show results only for two of the six selected windows.

The spatial standard deviations of Σ have been set to $\sigma_x = \sigma_y = 2^{\frac{2}{3}} \approx 1.59$

and the temporal one to $\sigma_t = 2.0$. Figure 2 shows the vectors $\mathbf{u}$ of equ. (3) projected into the image plane. The background of this image has been reduced in contrast and shifted to brighter gray values in order to offer a better contrast between the flow vectors and the original image. Note that the optical flow fields corresponding to the two vehicles are separated by a strip of zero optical flow corresponding to the road background despite the fact that the distance between the two car images is much less than the extent of the 7x7x9 up to 9x9x11 pixel filter masks. Note, in addition, the cues to the discontinuities in the optical flow fields of the left car caused by the three pixel wide post for the traffic sign and by the contour of the traffic sign itself.

Fig. 2. Enlargement of window No. 1 in Figure 1 showing two cars driving next to each other at approximately the same speed. The left car is partially hidden by a traffic sign post. The superimposed vectors depict the scaled eigenvector corresponding to the largest eigenvalue of $\widehat{\Sigma_0''}$ estimated at each pixel position.

We select the two larger eigenvalues σ_2^2 and σ_3^2 of $\widehat{\Sigma_0''}$ and define the confidence for an optic flow estimate by

$$conf(\mathbf{u}) = \frac{\left(\sigma_3^2 - \sigma_2^2\right)^2}{\left(\sigma_3^2 + \sigma_2^2\right)^2} \tag{10}$$

This fraction varies between 0 – in the extreme case where both eigenvalues are equal and thus the direction of the eigenvectors in a plane normal to $\mathbf{e}_1$ is undefined – and 1 for the other extreme where the smaller among these two eigenvalues vanishes. This latter case implies that the smallest eigenvalue vanishes, too (remember that $\widehat{\Sigma_0}$ should be positive semidefinite). The third eigenvector and thus the optic flow estimate derived from it will be the more reliable, the

closer $conf(\mathbf{u})$ approaches the value 1, since this expresses the fact that the last eigenvalue is much larger than the other two - see, too, [11].

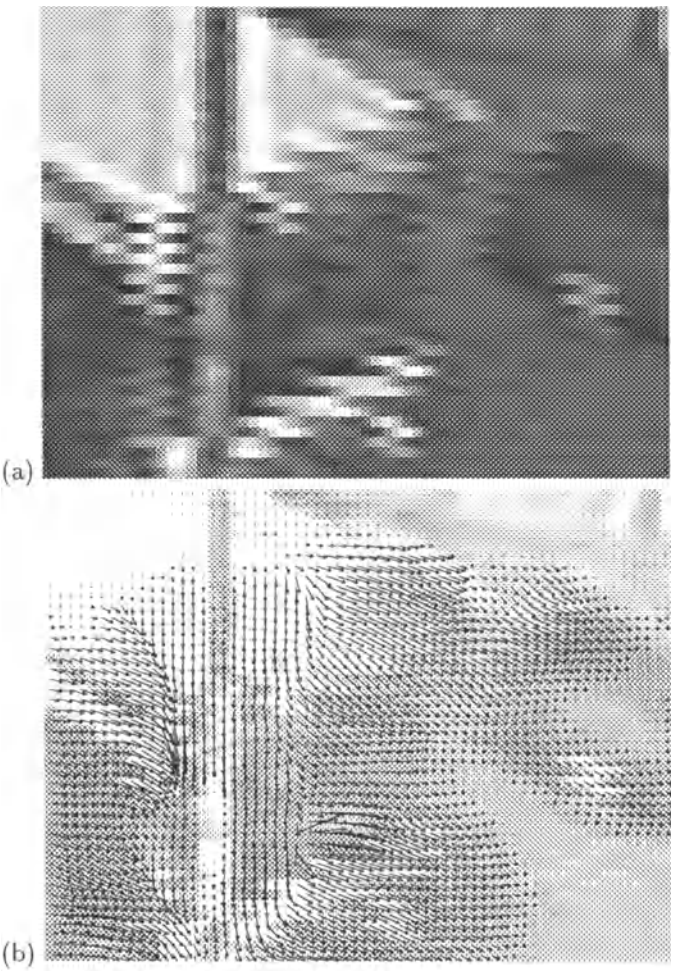


Fig. 3. (a) Enlargement of window No. 4 in Figure 1 showing the front part of the bus which is partially occluded by a street lamp. (b) Estimated optical flow vectors overlaid to (a).

Figures 3 and 4 show, too, results of window 4 from Figure 1 (comprising the front part of the bus and a faster moving car with relatively weak texture on the right hand side of the bus), with all vectors suppressed whose $conf(\mathbf{u})$ does not exceed a threshold of 0.01 . It can be seen that this suppresses the estimates of optic flow vectors in practically homogeneous areas of the bus or in those areas with predominantly straight line gray value transition fronts.

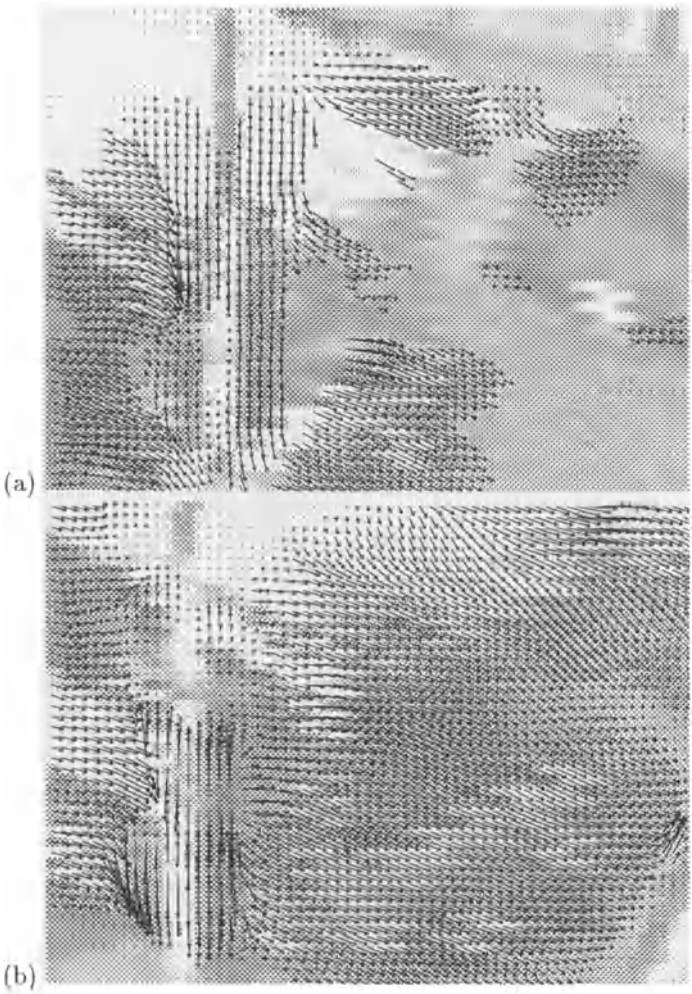


Fig. 4. (a) Analogous to Figure 3(b), but only those optical flow vectors are shown for which $conf(\mathbf{u}) > 0.01$. (b) Analogous to Figure 3(b), but from half-frame 33, i.e. $4\frac{1}{2}$ frames (180 msec) later.

The considerable magnification of the subimage shown in Figure 3 makes the effect of line jitter visible at the occluding edge of the lamp post. This sequence has been recorded with a high quality *tube* video camera. In order to demonstrate the effect which line jitter may have on estimates near optical flow discontinuities, Figure 5 shows a frame from an image sequence recorded by a CCD video camera (S-VHS). The image of the left support beam of the roof exhibits much less line jitter than the occluding lamp post in Figure 3. The transition area between optical flow vectors corresponding to the stationary occluding beam and the occluded moving car is restricted to a zone about 3–4 pixels wide. A similar

transition area can be observed between the left contour of the occluding petrol pump in the lower right corner of Figure 5 (b) and the occluded car behind it.

In view of the fact that each vector has been estimated independently from all surrounding ones, this illustrates the power of the approach described in this contribution. The fact that this approach facilitates a closed form sensitivity analysis of variations in the original assumptions is taken as an important aspect of attempts to move away from a purely experimental investigation of heuristics.

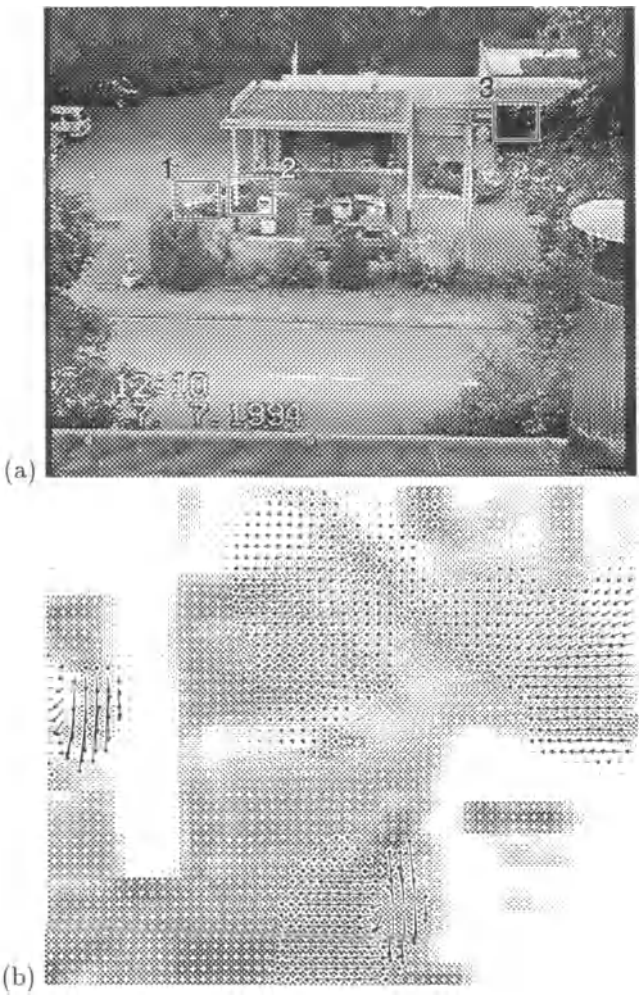


Fig. 5. (a) The 775th frame of an image sequence recorded at a small gas station in Karlsruhe, using a CCD video camera (S-VHS). An enlargement of the sub-image No. 2 is shown in (b), with estimated optical flow vectors superimposed. It can be seen that the image of the strongly contrasting left support beam of the roof exhibits much less line jitter than the occluding lamp post in Figure 3.

Acknowledgment

We thank H. Kollnig for many stimulating discussions about the topic of this contribution. H. Damm recorded the image sequence used in Figure 5. HHN extends his gratitude to G. Granlund and C.-F. Westin for making the latter's dissertation available.

This research has been partially supported by the European Community BRA project INSIGHT II.

References

1. J.L. Barron, D.J. Fleet, and S.S. Beauchemin: *Performance of Optical Flow Techniques.* International Journal of Computer Vision **12** (1994) 43-77
2. A. Gehrke: *Untersuchungen zur Segmentierung von optischen Flußfeldern.* Diplomarbeit, Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH), August 1995 (in German)
3. B.K.P. Horn: *Robot Vision.* The MIT Press, Cambridge/MA and London/UK 1986
4. B. Jähne: Analytical Studies of Low-Level Motion Estimators in Space-Time Images Using a Unified Filter Concept. Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '94, 21–23 June 1994, Seattle/WA, pp. 229–236
5. J.J. Koenderink, A. Kappers, and A. van Doorn: *Local Operations: The Embodiment of Geometry.* In: G.A. Orban and H.-H. Nagel (Eds.), Artificial and Biological Vision Systems, Springer-Verlag Berlin Heidelberg New York/NY 1992, pp. 1–23
6. H. Kollnig: *Ermittlung von Verkehrsgeschehen durch Bildfolgenauswertung.* Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe / Germany, February 1995. In: *Dissertationen zur Künstlichen Intelligenz* DISKI **88**, infix, Sankt Augustin / Germany 1995 (in German)
7. T. Lindeberg: Direct Estimation of Affine Image Deformations Using Visual Front-End Operations with Automatic Scale Selection. Proc. International Conference on Computer Vision ICCV '95, 20–23 June 1995, Cambridge/MA, pp. 134–141
8. H.-H. Nagel: *Analysis of Gradient Outer Product Tensor Fields.* Internal Report (Draft, 1995)
9. M. Otte: *Extraktion von linienförmigen Merkmalen und Ermittlung des optischen Flusses mit seinen Ableitungen aus Voll- und Halbbildfolgen.* Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe / Germany, February 1994. In: *Dissertationen zur Künstlichen Intelligenz* DISKI **63**, infix, Sankt Augustin / Germany 1994 (in German)
10. M. Otte and H.-H. Nagel: *Optical Flow Estimation: Advances and Comparisons.* Proc. Third European Conference on Computer Vision ECCV '94, Vol. I, Stockholm / Sweden, 2–6 May 1994, J.-O. Eklundh (Ed.), Lecture Notes in Computer Science **800**, Springer-Verlag Berlin Heidelberg New York/NY 1994, pp. 51–60. A revised and expanded version of this contribution will appear in Artificial Intelligence Journal (1995, in press).
11. C.-F. Westin: A Tensor Framework for Multidimensional Signal Processing. Linköping Studies in Science and Technology, Dissertations No. 348; Department of Electrical Engineering, Linköping University, S-58183 Linköping/Sweden (1994)
12. J. Zhang and H.-H. Nagel: *Texture-Based Segmentation of Road Images.* Proc. Intelligent Vehicles '94 Symposium, 24 – 26 October 1994, Paris/France, pp. 260–265

Combining the Hough Transform and Multiresolution MRF's for the Robust Motion Estimation

Mirosław Bober and Josef Kittler

Dept. of Electronic and Electrical Engineering,
University of Surrey,
Guildford GU2 5XH, United Kingdom.
email: m.bober@ee.surrey.ac.uk

Abstract

The paper presents a novel approach to the Robust Analysis of Complex Motion. It employs a low-level robust motion estimator, conceptually based on the Hough Transform, and uses Multiresolution Markov Random Fields for the global interpretation of the local, low-level estimates. Motion segmentation is performed in the front-end estimator, in parallel with the motion parameter estimation process. This significantly improves the accuracy of estimates, particularly in the vicinity of motion boundaries, facilitates the detection of such boundaries, and allows the use of larger regions, thus improving robustness. The measurements extracted from the sequence in the front-end estimator include displacement, the spatial derivatives of the displacement, confidence measures, and the location of motion boundaries. The measurements are then combined within the MRF framework, employing the supercoupling approach for fast convergence. The excellent performance, in terms of estimate accuracy, boundary detection and robustness is demonstrated on synthetic and real-word sequences.

1 Introduction

The problem of efficient and robust motion analysis has been attracting attention of researchers in Computer Vision for more then two decades. This acute interest has been motivated by many important applications which require motion analysis, and various solutions have been proposed. Several classes of algorithms exist: for the computation of optic flow, tracking, motion-based segmentation, camera motion compensation etc. However, many algorithms make restrictive assumptions (either explicitly or implicitly) regarding - for example - the type of motion involved, number of moving objects in the field of view, type of scene etc. This severely reduces their practical applicability and makes it difficult to select a suitable algorithm for the application at hand.

A broad class of algorithms performs motion analysis in two stages: optic flow estimation, followed by regularization (usually also involving discontinuity detection). The problem with such an approach is that the optic flow computed by many techniques is very noisy, in other words there are significant errors in local measurements of flow [1]. This is particularly the case with point-wise techniques based on the optic flow equation, but not necessarily restricted to

them. In [4] we have tested the robustness to image noise of several state-of-the art techniques, and found that even techniques using large temporal and spatial support (e.g. phase-based techniques) perform poorly in the presence of noise. It is usually left to the regularization procedure to remove the noise (smooth out the flow) and to detect the discontinuities. Clearly, these two demands are contradictory, and such approach is likely to fail when flow discontinuities are below the level of noise in measurements of displacement. Segmentation is further complicated by the fact that, in occluded or uncovered regions which are adjacent to the motion boundaries, and where the correspondence cannot be found at all, many techniques deliver random estimates. Finally, several techniques using a large temporal support will blur the estimates on the motion boundary thus making it impossible to find the motion edge.

The approach devised here uses a diametrically opposite strategy. The front-end estimator is designed to deliver robust estimate of motion and extract motion-related clues, but also involves strong regularization. This is achieved by using a region-based technique based directly on pixel grey-levels and by imposing constraints on motion of all pixels in the region. The technique uses a combination of Hough Transform with robust redescending kernels. It performs parallel motion estimation and segmentation, and therefore can cope with multiple moving patches within the region. The segmentation is based on the compensated frame-difference and the kernel shape is adjusted during the estimation. A four parameter motion model is used, thus the spatial derivatives of the flow are also estimated. The estimate confidence measures are based on the grey-level distribution within the region used for estimation. The technique delivers superior accuracy and copes very well with image noise and multiple motions.

The low-level estimates are refined within a Multiresolution Markov Random Field based framework. Several authors successfully used MRF's in the context of motion estimation and segmentation before [5, 2]. This is a very flexible framework, where different clues can be merged. The approach presented here has, however, two distinct features. Firstly, the regularization and discontinuity detection has been delegated to the front-end robust estimator. Secondly, the data used in the MRF framework includes, besides the displacement field, its spatial derivatives. The flow derivatives are extracted directly from the image sequence. This significantly improves the quality of final results, as the MRF labels are constrained not only by local measurements of the flow, but also by its derivatives. Thirdly, discontinuities are detected in the first stage and form a part of input-data to the MRF-based module. Finally, we applied a multiresolution optimization technique [8], based on so called supercoupling approach [7], which helps speedy convergence.

The rest of the paper is organized as follows. Section 2 describes briefly the low-level motion segmentation and estimation module. Several experiments testing the accuracy and robustness of the technique are presented. We also show how the confidence measures are computed and how the technique extracts motion boundaries. Section 3 describes the post-processing framework based on

Multiresolution Markov Random Fields. We also comment here on the optimization using the supercoupling approach. Experimental results are presented in Section 4. The summary of results and conclusions close the paper.

2 Problem Formulation

The following strategy is used to address the problem of motion analysis in the image plane. In the first stage, we use a low-level fast and robust algorithm to extract several motion related measurements (features) from the sequence. The features extracted are: the four parameter motion vector field, the motion estimate covariance matrix, a scalar number reflecting the probability of the motion measurements being correct, and the discontinuity labels. In the second stage, which can be viewed as a 'postprocessing', we use multidimensional Markov Random Fields, to find a globally coherent interpretation of our estimates. A block diagram of our framework is presented in Figure 1.

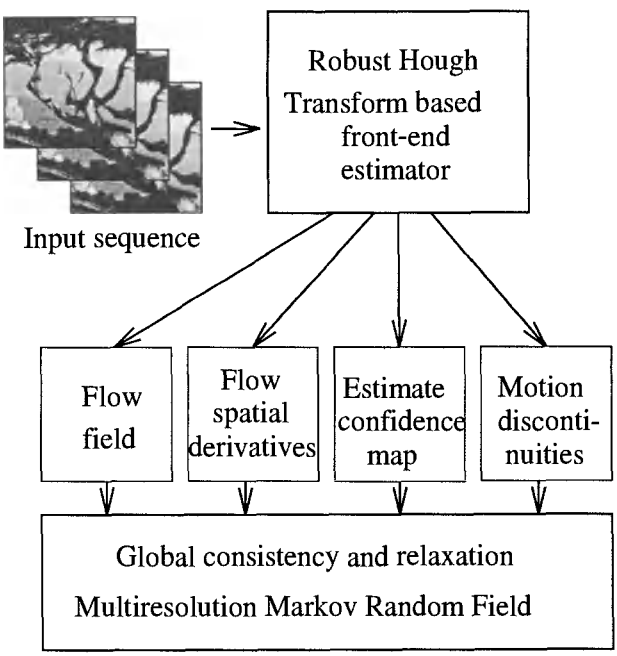


Fig. 1. Block diagram of the framework

2.1 Low-level Front-end Robust Estimator

The performance and features of the front-end estimator have a crucial impact on the overall performance of the framework. In terms of performance, one needs

high estimate accuracy (all motion parameters), estimate confidence measures and good robustness to noise and illumination changes. Also, it is desirable to have an initial indication of motion boundary, based directly on the raw grey-level data.

As the front-end estimator we selected a Local Robust Hough Transform Estimator (LRHT). Since its principles have been described in detail elsewhere [4, 3] we only briefly mention here several important features of the estimator.

The estimator uses only two consecutive frames and is region based - we use square patches. The main idea is that each pixel votes for a set of motion parameters, and the strength of vote depends on the value of the motion-compensated frame difference. The function that computes the strength of the vote from the motion-compensated difference is known as the 'kernel function'. A highly efficient, multiresolution optimization procedure is used to find the motion parameters that receive the strongest support. By setting a threshold on the motion-compensated frame difference computed for the determined set of motion parameters, the pixels the motion of which is not consistent with the estimated model can be segmented out. This feature is very important, because it makes it possible to base the estimation on larger regions without the danger of erroneous estimates in the vicinity of motion boundaries.

Another important advantage of our estimator is that it uses the affine motion model 1, which accurately describes translation, rotation, change of scale and shear. Our experience shows that, for small-to-medium region sizes (9x9 to 21x21 pixels) such model can describe motion with sufficiently good accuracy. In fact, in this framework a 4-parameter variant of the affine model (described by equation (2)) was used. For these models, pixel displacement in the image plane $\mathbf{d_p}$ is a function of the pixel position in the region $\Re$.

$$\mathbf{d_p} = (a_1 x + a_2 y + a_3, a_4 x + a_5 y + a_6) \tag{1}$$

$$\mathbf{d_p} = (a_1 x - a_2 y + a_3, a_2 x + a_1 y + a_4) \tag{2}$$

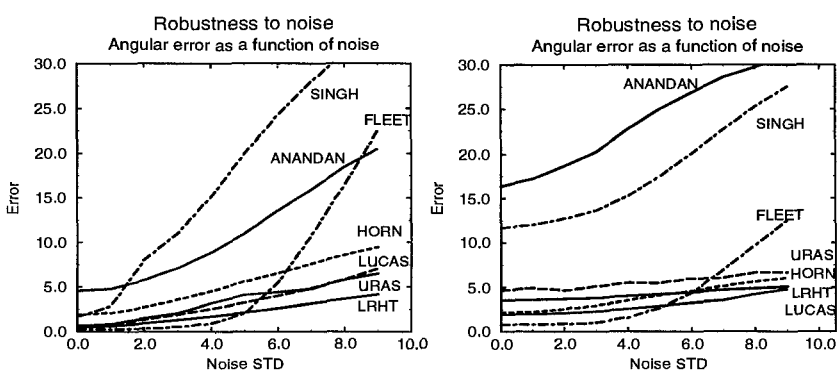
Parameters a_1, a_2 (or a_1, a_2, a_4, a_5 for the affine model) can also be interpreted as the spatial derivatives of the optical flow. Our estimator is capable of extracting accurately the spatial flow derivatives from the image.

Let us demonstrate the accuracy of the LRHT technique by comparing it to several state-of-the art techniques (Table 1). All the algorithms were run on the *Translating tree* sequence. Clearly, LRHT outperforms the other techniques, despite the fact that it uses a temporal support of only two frames! (The second best, phase-based technique by Fleet and Jepson, uses 31 frames). The angular error of 0.3 degrees (as defined in [1]) corresponds to a relative flow magnitude error of 0.05 and flow angular error below 1 deg ! It can be also seen that our technique provides very dense flows. The next graph also shows the robustness of several estimators. A zero-mean Gaussian noise was added to each frame in the sequence, and the deterioration of the estimate accuracy was computed as a function of noise standard deviation. Again, the accuracy of the estimate computed by competing techniques seems to deteriorate very quickly, while LRHT gives a reliable motion field even for a noisy sequence. In our implementation

Technique	Average Error [deg]	Standard Deviation	Flow Density [%]
Horn and Schunck	33.40	16.46	100%
Uras -unthresholded	12.48	17.52	100%
Anandan	4.54	2.98	100%
LRHT-unthresholded	**0.31**	**0.22**	**100%**
Heeger	4.79	2.39	13.8%
Lucas and Kanade	1.75	1.43	40.8%
Fleet and Jepson	0.36	0.41	76%
LRHT-unthresholded	**0.31**	**0.22**	**100%**
LRHT-thresholded	**0.27**	**0.17**	**79.1%**

Table 1. A comparison of the estimate accuracy for the **Translating Tree** sequence

regions of rectangular shape are used, and the reference coordinate system is placed in the center of each block. However regions could have any arbitrary shape and even overlap. In the experiments presented in this subsection, non-overlapping blocks of 15×15 pixels were used. Each frame was filtered with a low-pass Gaussian filter with $\sigma = 2$. Partially overlapping blocks give even better accuracy, but require more computation.



(a) Translating Tree sequence (b) Diverging Tree sequence

Fig. 2. Estimate accuracy as a function of noise for the the Translating and Diverging Tree sequence

At each pixel position, we extract four motion parameters: m_1, m_2, m_3, m_4

which form a 4-dimensional motion vector $\mathbf{M}$. Components m_1, m_2 correspond to the spatial derivatives of the flow and m_3, m_4 are the x and y components of the flow respectively. We compute the estimate covariance matrix $\mathbf{C}$, which depends on the texture distribution within the block used for estimation. To simplify the problem, we assume that the confidence in estimates m_3, m_4 is independent on confidence in m_1, m_2 and that confidence in m_1 is independent on confidence in m_2. The aperture problems will be reflected in the matrix $\mathbf{C}$.

$$\mathbf{C} = \begin{vmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & c_{33} & c_{34} \\ 0 & 0 & c_{34} & c_{44} \end{vmatrix} \tag{3}$$

We also monitor the number of outliers detected by the Robust Kernel in order to find the regions where motion boundaries are present. The outlier map, defined as a ratio of number of pixel-outliers to the number of pixels within a block, is computed and used in the postprocessing stage.

2.2 Global Post-processing of Local Measurements Using Markov Random Fields

Markov Random Fields are used at this stage to impose a global consistency on the measurements extracted by the low-level estimation module. Markov Random Fields have been used before for motion analysis, but our framework differs in several important aspects.

- ◇ In existing approaches, the recovery of motion boundaries is solely based on detecting the discontinuities in the (usually noisy) input flow. In [5, 8] grey-level edges were also used as supporting evidence. In our framework, not only optic flow, but also its derivatives, are extracted directly from the sequence. The process of detection of discontinuities is therefore greatly simplified. In principle it should be also possible to detect discontinuities in the first derivatives of the flow. Moreover, the regularization used in the first stage gives a more accurate initial estimate and provides a segmentation map.
- ◇ Better motion segmentation and a more accurate initial estimate contribute to the quality of the final motion estimate. It will be shown that excellent accuracy is achieved.
- ◇ The use of quasi-affine motion model in the MRF framework helps to cope with complex motions [2].
- ◇ In order to improve convergence and simplify the selection of model parameters we applied the supercoupling optimization approach. The idea is that the data and model parameters are mapped between resolutions in such a way that the position of the global minimum in the parameter space is preserved [8].

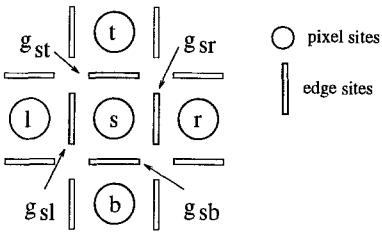


Fig. 3. The neighborhood system N used

Let us define the computational framework. It is based on Markov Random Fields, with the embedded line process. Our objective it to recover the motion parameters $\mathbf{A_i}$ at each pixel location i and motion discontinuities labels Γ at inter-pixel locations (Figure 3). The data extracted by the front-end estimator includes: motion parameters recovered from the sequence $\mathbf{M}$, the confidence matrix $\mathbf{C}$ and the discontinuity map $\mathbf{G}$. Vector $\mathbf{G_s}$ is defined as $\mathbf{G_s} = (g_{st}, g_{sr}, g_{sb}, g_{sl})$, where g_{si} is the motion discontinuity estimate between sites s and i ($i \in \{t, r, b, l\}$). Let $\mathcal{A}$ denotes a set of data and labels $\mathcal{A} = \{\mathbf{V}, \mathbf{C}, \mathbf{G}, \mathbf{A}, \Gamma\}$. The probability distribution of MRF can be expressed as:

$$p(\mathcal{A}) = \frac{1}{Z} e^{-\frac{1}{T}U(\mathcal{A})} \tag{4}$$

$$U(\mathcal{A}) = \sum_{s \in image} \alpha_1 V_{1,s} + \alpha_2 V_{2,s} + \alpha_3 V_{3,s} \tag{5}$$

The normalizing constant Z is the partition function and parameter T is the "temperature" constant. The cost function $U(\mathcal{A})$ is the weighted sum of three terms, each expressing a constraint on the estimated labels $\mathbf{A}, \Gamma$.

◇ The first term, V_1, constraints the estimate $\mathbf{A}$ to agree with low-level motion estimate $\mathbf{M}$. The penalty depends on the confidence measure $\mathbf{C}$.

$$V_{1,s} = (\mathbf{A_s} - \mathbf{M_s})\mathbf{C_s}(\mathbf{A_s} - \mathbf{M_s})^{\mathbf{T}} \tag{6}$$

◇ The second constraint relates the discontinuity labels Γ recovered and discontinuities detected by the front-end estimator $\mathbf{G}$. It penalizes the configurations where discontinuity label is assigned at locations where the LRHT estimator shows no (little) outliers.

$$V_{2,s} = (\mathbf{1} - \mathbf{G_s}) \circ \Gamma_\mathbf{s} \tag{7}$$

◇ Finally, the third term imposes smoothness constraint on the flow and its derivatives:

$$V_{3,s} = (\mathbf{1} - \Gamma_\mathbf{s}) \circ \sum_{k \in N_s} \mathbf{O(k)}(\mathbf{A_s} - \mathbf{A_{sk}})\mathbf{K}(\mathbf{A_s} - \mathbf{A_{sk}})^{\mathbf{T}} \tag{8}$$

where $\mathbf{O}(k) = (o_t, o_r, o_b, o_l)$ and all elements $\{o_i, i \neq k\}$ are equal 0 except for o_k which equals 1. Note, that the smoothness constraint between neighbors s and k is inhibited by the placement of motion boundary $\gamma_{sk} = 1$. The diagonal matrix $\mathbf{K}$ has $k_{11} = k_{22} = k$ and $k_{33} = k_{44} = 1$ where k is a constant scaling parameter. It is introduced in order to take into consideration the different nature of the two sets, $\{a_{1,s}, a_{2,s}\}$ and $\{a_{3,s}, a_{4,s}\}$ of the motion parameters. Parameters $a_{1,s}$, $a_{2,s}$ have dimensions of linear velocity, and parameters $a_{3,s}$, $a_{4,s}$ have dimensions of angular velocity.

Finding the MAP configuration of the labels has now been reduced to an optimization problem. We use the supercoupling approach, which is described in detail in [8].

3 Experimental Results

We present here three experiments, each designed to show a different aspect of the presented framework, each illustrating its improved performance.

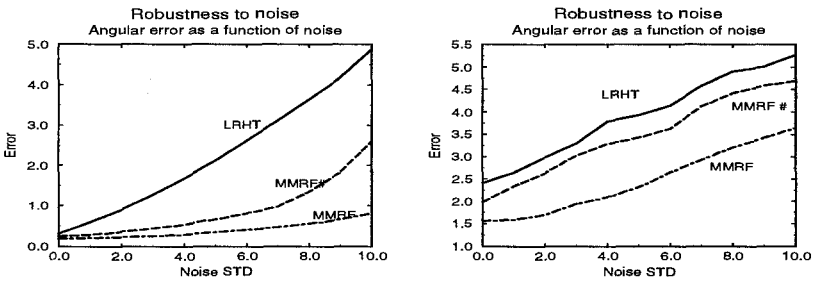
In the first experiment, we compare the estimate accuracy between the full version and a variant which uses only optical flow, but not the derivatives. Again, the algorithms were applied to translating and diverging tree sequences. The mask size was 15×15 pixels, and overlapping blocks were used. The results are summarized in Table 2. It is clear that the use of flow derivatives extracted from the sequence improves the accuracy of the estimate and reduces its standard deviation.

In the next experiment we compare the robustness of the estimate to Gaussian noise. The performance is shown on Figure 4. The benefit of using flow derivatives as an input for the postprocessing stage is again clear - the robustness of our framework is significantly increased, especially in case of affine motion (The Diverging Tree sequence).

Finally, we show the performance of the algorithm on a real-word sequence with multiple motions called **Mobile & Calendar**. Figure 5(a) shows the reference frame from this sequence. The sequence depicts a scene where the wall calendar moves in the top-right direction, the toy-train moves to the left and the camera is panning to the left (therefore the background moves to the right). There is also a small toy moving in the left-bottom corner of the image and a

Technique	Average Angular Error [deg]	Standard Deviation	Flow Density [%]
Bober & Kittler **LRHT**	0.31	0.22	100%
Bober & Kittler **MMRF♣**	0.25	0.15	100%
Bober & Kittler **MMRF**	0.22	0.12	100%

Table 2. A comparison of the estimate accuracy for the **Translating Tree** sequence, LRHT, MMRF♣ (Multiresolution MRF based framework without flow derivatives), and fully functional MMRF



(a) Translating Tree sequence (b) Diverging Tree sequence

Fig. 4. Estimate accuracy as a function of noise for LRHT, MMRF♣ (MMRF framework without flow derivatives), and fully functional MMRF

ball rolling in front of the train. The optic flow recovered by the LRHT estimator and its components (represented as grey-levels, with grey-level 127 corresponding to no motion) are shown in Figures 5(b), 5(c),5(d) respectively. Figure 5(e) shows the motion boundaries recovered by the MMRF framework. Most of the discontinuities in the flow were correctly recovered, including small elements of the moving toy. However, the ball was not segmented properly. In our view, the segmentation results can be further improved, by imposing a stronger constraints on the spatial/temporal continuity of motion boundaries.

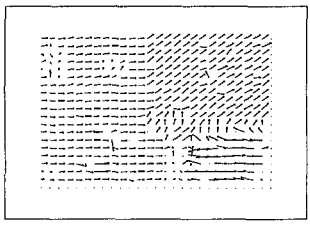
4 Conclusions

We have presented a novel approach to the analysis of complex motion. It employs a low-level Robust motion estimator, conceptually based on the Hough Transform, and uses Multiresolution Markov Random Fields for the post-processing. Regularization and motion segmentation is performed in the front-end estimator, in parallel with the motion parameter estimation process. This significantly improves the accuracy of estimates, particularly in the vicinity of motion boundaries, facilitates the detection of such boundaries, and allows the use of larger regions thus significantly improving robustness. The measurements extracted from the sequence in the front-end estimator include displacement, the spatial derivatives of the displacement, confidence measures, and the location of motion boundaries. The measurements are then combined within the MRF framework, employing the supercoupling approach for fast convergence. The excellent performance, in terms of estimate accuracy, boundary detection and robustness is demonstrated on synthetic and real-world sequences.

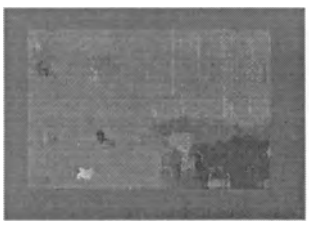
Acknowledgments

This work was partially supported by the Engineering and Physical Sciences Research Council grants GR/J52563 and GR/K08598.

(a) Reference frame

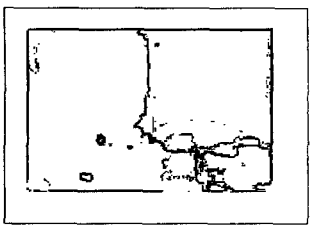


(b) Optic flow extracted by LRHT



(c) Component velocity - x direction

(d) Component velocity - y direction



(e) Final Motion Boundaries

Fig. 5. Results for **Mobile & Calendar** sequence

References

1. J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A Burkitt. Performance of optical flow techniques. In *Proceedings CVPR'92 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL.,1992)*, pages 236–242, Washington, DC., 1992. Computer Society Press.
2. M. Bober and J. Kittler. General motion estimation and segmentation. In *Proceedings Conf Pattern Recognition and Image Analysis, Valencia, Spain, September 1992*, pages 10–18, 1992.
3. M. Bober and J. Kittler. Estimation of general multimodal motion: an approach based on robust statistics and Hough transform. *Image and Vision Computing*, 12(12):661–668, 1994.
4. M. Bober and J. Kittler. Robust motion analysis. In *Proceedings, CVPR '94 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition), Seattle, June 20-24, 1994*, pages 947–952. IEEE Computer Society Press, 1994.
5. F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1217–1232, 1993.
6. *Proceedings, International Conference on Pattern Recognition, Jerusalem, October 1994*. IEEE Computer Society Press, 1994.
7. G. K. Nicholls and M. Petrou. On multiresolution image restoration. In ICPR1994 [6], pages 63–67.
8. M. Petrou, M. Bober, and J. Kittler. Multiresolution motion segmentation. In ICPR1994 [6], pages 379–383.

Direct Model-Based Image Motion Segmentation for Dynamic Scene Analysis

Jean-Marc Odobez and Patrick Bouthemy *

IRISA/INRIA, Campus universitaire de Beaulieu
35042 Rennes Cedex, France
e-mail : odobez@irisa.fr bouthemy@irisa.fr

Abstract. Analysing the dynamic content of a scene observed by a mobile camera often requires the segmentation of each image of the sequence into region entities of apparent homogeneous motion. To each region is associated a 2D polynomial model (e.g., an affine one) able to describe at each location the underlying 2D "true" motion with a predefined precision η. Thanks to the use of a multiresolution robust estimator [1] to compute the motion models, the determination of the boundaries between the different regions, which is stated as a statistical regularization based on multiscale Markov Random Field (MRF) models, can be achieved in one pass only. This avoids the time consuming iterations between motion estimation and boundary identification that are encountered in almost all other motion-segmentation schemes (for instance [2, 3, 4]). We explicitly detect areas where the error between the underlying motion and the one given by the estimated models is not whithin the precision η. This allows us to handle the appearance of new objects in the scene. We have performed numerous experiments with real indoor and outdoor image sequences which demonstrate the efficiency of the method.

1 Introduction

The apparent motion field existing in a monocular image sequence acquired by a moving camera contains a rich source of information about camera motion and the composition of the viewed scene. Therefore, in the past years, substantial research efforts have been devoted to the estimation of the 2D motion field (dense or not) followed by structure-from-motion or egomotion computation. However, the predominance of this approach led to the under-exploration of other important issues related to the analysis of time-varying imagery, where the interpretation of the 2D motion in terms of scene content can be put at an earlier level than egomotion or 3D structure computation. Motion-based segmentation [2, 3, 4, 6], that is, the partitioning of the image in visually important

* This study was supported in part by the French Ministry of Research in the context of the GDR-PRC "Man-Machine Communications"(Vision research program, MRT contract 91S269), and by "Région Bretagne" (Brittany Council) through a contribution to student grant.

regions that are homogeneous according to a given motion criterion, is one of these. Such a segmentation gives a structured representation of the scene. Indeed, the regions thus discriminated, sometimes referred to as layers [4], reveal the different 3D motion, depth or orientation of the 3D surfaces whose projection they are. The need for motion segmentation appears as soon as the camera is moving and/or the scene contains several moving objects. A good segmentation, by localizing the motion discontinuities not only in the image domain but also in time through temporal integration, can contribute to the improvement of numerous tasks. It is generally preferable to first segment the image of the scene into its different kinematic components in order to estimate or identify a given quantity on homogeneous spatio-temporal data. This is the case, among others, when we want to estimate a dense motion field [6]; to compute time-to-collision [7] or similarly to avoid obstacles [8]; to qualitatively [3] or even quantitatively [9] interpret the 3D dynamical scene content. In this paper, we present a new motion-based segmentation method, which could be used as a first step or in combination with previously mentioned applications.
In section 2, we give the general features and assets of our approach, while in section 3, we describe the different steps of the motion-based segmentation algorithm that we propose. Section 4 reports results, obtained on real image sequences, which validate our method.

2 General Features

Our motion-based segmentation algorithm relies on the use of 2D parametric motion models, the robust estimation of these models, and the introduction of multiscale Markov Random Field (MRF) models. The goal of the segmentation is to jointly estimate the motion models $\{(\Theta_k)_t^{t+1}\}$ between time t and time $t+1$ in each delimited region $R_k(t)$, $k \in \{1, \ldots, N_r\}$, and the associated partition into regions represented by a label field $e(t)$ at time t[2], whose labels are in the set $\{1, \ldots, N_r\}$. N_r stands for the number of regions in the image, and has to be estimated on-line also. Though any 2D polynomial motion model could be considered, we mainly dealt with the affine one. With this model, the displacement $\mathbf{d}_{\Theta_k}$ at point $s = (x, y)$ in region k is described by:

$$\mathbf{d}_{\Theta_k}(s) = \begin{pmatrix} a_1^k + a_2^k x + a_3^k y \\ a_4^k + a_5^k x + a_6^k y \end{pmatrix} \text{ with } \Theta_k = (a_i^k)_{i=1\ldots6} \tag{1}$$

The main requirements and motivations behind the design of a direct model-based segmentation algorithm are the following:

1. we want to segment the *underlying motion field*. Therefore, rather than employing the usual displaced frame difference:

$$DFD(s,k) = I_{t+1}(s + \mathbf{d}_{\Theta_k}(s)) - I_t(s)$$

[2] We will drop the time indexes when there is no ambiguity.

as a motion criterion, we look for local motion measurements that quantitatively indicate whether a modeled flow at site s, $\mathbf{d}_{\Theta_k}(s)$, constitutes a good approximation of the underlying true flow at this pixel, $\mathbf{d}_{\text{true}}(s)$. More precisely, if η is a predefined constant, we would like to have in each region R_k:

$$\forall s \in R_k, \quad \|\Delta\mathbf{d}(s,k)\| \leq \eta, \tag{2}$$
$$\text{with} \quad \Delta\mathbf{d}(s,k) = \mathbf{d}_{\text{true}}(s) - \mathbf{d}_{\widehat{\Theta}_k}(s)$$

However, we want to avoid the use of a previously estimated flow field, as done in [4, 10], since such a flow field might be very noisy if estimated locally, or blurred at motion boundaries if a global regularization technique is employed. Furthermore, it is usually time consuming to compute optical flow field. Hence, we only resort to parametric motion model estimated from spatio-temporal intensity derivatives, and to partial motion measurements to locally assess the validity of the estimated model according to (2). This is the first meaning of the term "direct" used to qualify our method.

2.most approaches to segmentation generally proceed in two steps that are iterated until convergence. The first one consists in estimating the motion models given the current partition; the second one in determining the optimal partition, the motion models being kept unchanged [4, 3, 6]. This optimization procedure happens to be computationally expensive, especially when stochastic minimization is utilized [2]. To reduce this complexity, these successive iterations should be avoided. In our case, this is achieved thanks to the use of a robust multiresolution motion estimator described in [1], as explained below. This is the second motivation to present our motion segmentation method as a "direct" one. Let us note that Ayer et *al* [5] also use robust estimation for segmentation purpose. However, as the motion model estimation is based on a least-median-square technique along with a temporal decomposition, and the segmentation relies on a merging procedure from an initial spatial segmentation, their algorithm is far more complex than ours.

3 Segmentation Algorithm

The determination of the segmentation map at time t, given the segmentation map at time $t-1$, involve four basic steps: the prediction of an initial segmentation map, the estimation of the motion models, the updating of the predicted map given these computed motion models, and finally the detection of new regions. We describe these steps hereafter.
The search for the motion-based segmentation corresponding to the two first images of the sequence is conducted in the same way, starting with an initial segmentation map composed of one single region.

3.1 Prediction

The prediction of the partition at time t, denoted $\tilde{e}(t)$, is determined using the segmentation map along with the estimated motion models obtained at time

$t-1$. This (along with step 3) allows us to supply a coherent labeling of the same moving scene element in the successive partitions over time. This is of great importance for further analysis processing steps, which may require a temporal tracking of regions over time and/or an estimation of the temporal derivatives of motion parameters (see [7] for instance, where time-to-collision maps are derived). More precisely, the label k in $\hat{e}(t-1)$ at each pixel s is assigned to each point on the grid around $s + \mathbf{d}_{(\widehat{\Theta}_k)_{t-1}^{t}}(s)$ in $\tilde{e}(t)$. Pixels that receive no label are given a special label, as well as pixels that get several labels. They respectively correspond to uncovered regions between $t-1$ and t, and to occlusion areas. This allows us to properly take into account these situations, which is of importance when dealing with real complex scenes.

3.2 Robust Motion Estimation

The motion models $(\Theta_k)_t^{t+1}$ between frame at time t, I_t and frame at time $t+1$, I_{t+1}, are estimated using the initial partition $\tilde{e}(t)$. Since we have designed a robust estimator [1], an imprecise predicted map or the appearance of new objects do not perturb this estimation process. Therefore, parameters $(\Theta_k)_t^{t+1}$ are computed only once. More precisely, our estimation method exploits a multiresolution framework and an incremental scheme based on the Gauss-Newton method. It minimizes an M-estimator criterion involving a hard-redescending function to ensure the goal of robustness. Each $(\Theta_k)_t^{t+1}$ is estimated as follows [1]:

$$(\hat{\Theta}_k)_t^{t+1} = \underset{\Theta_k}{\operatorname{argmin}} \sum_{s \in \tilde{R}_k(t)} \rho\left(\mathrm{DFD}(s,k)\right) \tag{3}$$

where $\rho(x)$ is the Tuckey's function, bounded for high values of x, and $\tilde{R}_k(t)$ is the kth predicted region support.

3.3 Updating of the Predicted Partition

The updating of the partition $e(t)$ using the predicted one $\tilde{e}(t)$ and the estimated models $(\hat{\Theta}_k)_t^{t+1}$, is achieved through a statistical regularization approach based on multiscale MRF. We adopt a Bayesian MAP criterion, which, due to the use of MRF models, leads to the minimization of an energy function $U(e, o, \tilde{e})$, where o is the field of observations and is composed of the images I_t and I_{t+1}. We have established an energy function comprising three terms:

$$U(e, o, \tilde{e}) = U_1(e, o) + U_2(e) + U_3(e, \tilde{e}) \tag{4}$$

• We have paid particular attention to the data-driven term U_1, which expresses the adequacy between the labels and the observations. To fulfill the first requirement mentioned above, we use partial local *motion* measurements $o(s,k)$ that can be *straightforwardly* computed from the images.
Examination of the structure of the $DFD(s,k)$ and experience shows that the DFD is essentially driven by the spatial intensity gradient term rather than by the presence of high or low underlying residual motion $\Delta\mathbf{d}(s,k)$. Therefore, the

normal residual flow $\Delta d_n(s,k) = |DFD(s,k)|/||\nabla I(s)||$ is a more appropriate measurement. It suffers, however, from being noisy in uniform areas, as well as from the aperture problem. To alleviate these difficulties, we have adopted as a measurement a weighted average of Δd_n over a 3×3 neighborhood $N(s)$ of site s. As weights, we take $||\nabla I||^2$, since the spatial image gradient is often considered as a good indicator of the relevance of the normal flow measurements. Thus, we have:

$$o_s(k) = \frac{\sum_{p\in N(s)} |DFD(p,k)| \times ||\nabla I(p)||}{\max(m, \sum_{p\in N(s)} ||\nabla I(p)||^2)} \tag{5}$$

where m is a predetermined positive constant related to the noise level in uniform areas. Associated with this measure, we can derive [11] two bounds $l(s)$ and $L(s)$, which only depend on the local distribution of the directions of the spatial intensity gradient and on the preset parameter η (see (2)). Then, we can show that:

$$\begin{cases} o_s(k) < l_s \Rightarrow ||\Delta \mathbf{d}(s,k)|| < \eta \\ o_s(k) > L_s \Rightarrow ||\Delta \mathbf{d}(s,k)|| > \eta \end{cases}$$

The potentials involved in the energy term U_1 reflect the "energetic translation" of the above inequalities:

$$U_1(e,o) = \sum_{s\in S} \alpha.F_s.V_1(e_s,o) \text{ with } F_s = A_{G,1}(||\nabla I(s)||)$$

$$\text{and } V_1(e_s,o) = A_{l_s,2}(o_s(e_s)) - (1 - A_{L_s,2}(o_s(e_s))) \tag{6}$$

where $A_{tr,r}(x)$ is a smoother version of a step edge. It is an increasing function from 0 to 1, such that the step occurs at tr ($A_{tr,r}(tr) = 0.5$) and the slope of the transition is controlled by r ($\mathrm{d}A_{tr,r}/\mathrm{d}x(tr) = r$). We use the normalized arctangent: $\frac{1}{\pi}\arctan(r\pi(x-tr)) + 0.5$ instead of a sigmoide, because it reaches values 0 or 1 less rapidly. A site with low image gradient usually carries poor and unreliable information about the adequacy of a given motion model. Therefore, the role of the damping factor F_s is to reduce the amplitude of the energy term V_1 provided by the observations at such a site (approximately characterized by $||\nabla I(s)|| < G$), which *conversely* increases the relative contributions of the regularization terms.

The bounds and the above inequalities allow us to take into account the well known "aperture problem". More precisely, they enable to tell, *given the local intensity structure*, which motion model is "adequate" (i.e., describes the underlying true motion with precision η), and which is not. This is reflected in the potential functions, as illustrated in Figure 1 where we have drawn the potential functions for two representative cases. Let us note that a very low potential value should indicate that the residual motion magnitude is below a given predefined value η. In the straight edge case, due to the aperture problem, an observation, however low it is, cannot tell with certainty whether the residual motion magnitude is below η. Therefore, the energy does not fall as low as in other cases, like in the corner case, where the aperture problem does not arise.

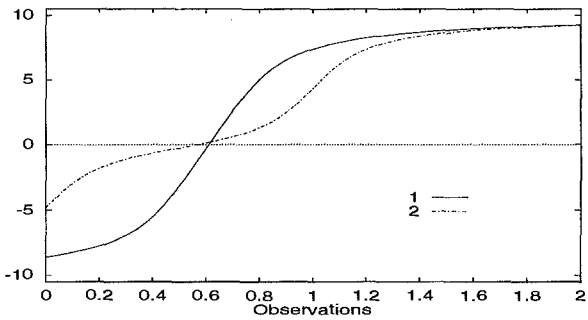


Fig. 1. Potential function V_1 in two extreme cases: case of a site located on a corner (curve 1), and on straight edge (curve 2).

When there exists an ambiguity (for instance, if there is not enough structure, like in uniform areas, or if several motion models are convenient,...), energy terms U_2 and U_3 that are described below introduce the contextual information necessary to remove it and to perform a correct labeling.

Let us note that U_1 has been easily extended to account for areas at time t that are occluded at time $t+1$. To correctly label these areas, the most intuitive way consists in looking towards the past. Therefore, in our scheme, in addition to motion estimates from time t to time $t+1$, $(\hat{\Theta}_k)_t^{t+1}$, we have also considered motion models from t to $t-1$, $(\Theta_k)_t^{t-1}$. Then[3], we can define motion measurements $o'_s(k)$ involving these models and the images I_t and I_{t-1}, in a way similar to the definition of $o_s(k)$. When the right label k is assigned to s, at least one of the measure $o_s(k)$ or $o'_s(k)$ should be small, whereas with a wrong label, both should be high. Thus, the energy term U_1 is simply modified by replacing $o_s(e_s)$ in (6) by the minimum of the motion measures $o_s(e_s)$ and $o'_s(e_s)$. This simple way of handling occlusion is "transparent", in the sense that it does not tell us directly which layer is passing in front of the other. However, once the segmentation maps has been estimated, this information could easily be retrieved in a complementary phase (and could be used in the prediction step).

• Energy term U_2 accounts for the expected spatial properties (homogeneity) of the label field. It has the following expression (usual one):

$$U_2(e) = \sum_{(s,t)\in\mathcal{C}} \beta_d(1 - \delta_{e_s=e_t})$$

where $\mathcal{C}$ represents the set of cliques of two elements associated to a second order neighbourhood system ν, and δ is the Kronecker function.

• U_3 favours the conservation of labels over time (except in occluded and uncovered regions between I_{t-1} and I_t where no label is favoured), if the newly estimated motion models $(\hat{\Theta}_k)_t^{t+1}$ are still able to correctly describe the motion

[3] Note that the affine model, $(\Theta_k)_t^{t-1}$ can be directly obtained from $(\hat{\Theta}_k)_{t-1}^{t}$; when using more complicated motion models (quadratic polynomial for instance), $(\Theta_k)_t^{t-1}$ must be estimated in the robust estimation step.

inside their corresponding regions. It is given by:

$$U_3(e, \tilde{e}) = \sum_{s \in S} F_s \times \beta_{dt}(1 - \delta_{e_s = \tilde{e}_s})$$

where δ is again the Kronecker function. Let us point out that this energy term also plays a second role. Let us suppose that a new region l is created (see next section) in a sub-area inside the region k if the motion estimate $\hat{\Theta}_k$ is no more adequate to describe the motion in this area. This new region l, of course not existing in $\tilde{e}$, has therefore to bring a significant gain in the description of motion in this area to compensate for the penalty introduced by the potential V_3. As the possible gain is weighted by F_s in U_1, the penalty term V_3 must also be weighted by F_s.

The global minimization of the energy function is performed using a multi-scale approach, [12], where at a given scale the solution is computed with the Highest Confidence First minimization procedure [13].

In this 3rd step of the algorithm, the areas labeled "occluded" or "uncovered" in the prediction step (section 3.1) are assigned the labels that suit them. The updating of the boundaries between two regions k and k', according to the new motion estimates, is also performed. Thus, at the end of this step, the segmentation into regions of homogeneous motion is achieved using the number of motion models intervening in the segmentation at the previous instant. The purpose of the step that follows is therefore to test: a) if new mobile objects have appeared in the scene; b) if the current number of motion models is still enough to provide a good description of the apparent motion in the image (e.g., it tests whether there is an increase in the complexity of the apparent motion field).

3.4 Detection of new Regions

Within each region k, sub-areas whose motion do not conform (in the sense of (2)) to the estimated motion model $\hat{\Theta}_k$ are detected. This is achieved using a scheme similar to the one described in [14], which was concerned with the detection of moving regions not conforming with the global motion model estimated in the whole image. Briefly, this detection phase uses a statistical regularization leading to the minimization of an energy function U_d similar to (4). In particular, the data driven term in U_d can be directly computed from (6) at no cost, leading to an extremely fast detection procedure. The significant connected components are extracted from the set of all those sub-areas where the existing motion models are not valid, and the number of regions N_r is updated accordingly. If there is no new region, the final partition at time t is that obtained at the end of the last relaxation iteration performed at step 3. Otherwise, the motion models, *in the newly created regions only*, are estimated still using the multiresolution robust estimator, and the partition is updated again according to the relaxation scheme described in step 3. It is important to note here, that the relaxation now involves only few computations, since it is only concerned with the adjustment of the boundaries of the created regions.

4 Results and Conclusion

This algorithm has been validated on many real sequences comprising indoor and outdoor scenes. Here, we give results obtained on the "van" and the "garden" sequences (Fig. 3 and 2 respectively), where all parameters related to the segmentation algorithm are identical ($\alpha = 200$, $\beta_d = 20$, $\beta_{dt} = 40$, $G = 1$), except two. The first one is the precision parameter η mentioned in (2); the other one is the parameter m, related to the amount of noise in the image. Their values are given in the captions.

In the garden sequence, the camera is translating to the right, while the scene is static. Several motion models are needed to structure this scene into its different layers. Indeed, it is well known that an affine model is a good approximation of the motion of 3D planar surfaces or shallow objects. The garden sequence illustrates nicely this property. The ground as well as the houses in the background can be considered as planar surfaces with different orientations, while the two regions in the tree can be considered as shallow objects. Boundaries between regions or layers correspond not only to motion or depth discontinuities, as shown by the estimated modeled flow field displayed in Fig. 2b, but also as said earlier, to orientation discontinuities.

In the van sequence (Fig 3a-d), the scene takes place at a crossroad. A white van, followed by a black car driving at almost the same speed, is coming from the left of the image and going to the right. A white car is coming from the opposite direction, driving slightly towards the camera. The results demonstrate the accuracy of our algorithm in determining the motion dicontinuities, even when the white car is getting occluded (Fig 3b). The improvement of the algorithm obtained by modifying the energy term U_1, in order to use past information, is at work here. Fig. 3c shows the results at time t_{53}. At this instant, the size of the white car image projection is very small. Thus, the creation of a new region to account for this appearing car is postponed to the next instant, as shown by Fig. 3d.

In this paper, we have described an algorithm to directly segment the motion field without requiring its estimation. The segmentation results we have obtained in all our experiments are very consistent. They could be used for different applications as mentioned in the introduction. Topics for future research include the use of more complex motion models in our scheme, as well as the development of an algorithm to group the different segmented regions that are undergoing the same 3D rigid motion.

References

1. J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *To appear in Int. Jal of Visual Communication and Image Representation*, September 1995.
2. D.W. Murray and H. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.9, No.2:220–228, March 1987.

3. P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. Journal of Computer Vision*, Vol.10, No 2:157–182, April 1993.
4. E.H. Adelson and J.Y.A Wang. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3(5):625–638, September 1994.
5. S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Proc. of the 3rd ECCV*, volume 2, pages 316–327, Stockholm, May 1994.
6. M.M. Chang, M.I. Sezan, and A.M. Tekalp. An algorithm for simultaneous motion estimation and scene segmentation. In *Proc. of ICCASP 94*, volume 5, pages 221–224, Adelaide, Australia, June 1994.
7. F. Meyer and P. Bouthemy. Estimation of time-to-collision maps from first order motion models and normal flows. In *Proc. 11th Intern. Conf. on Pattern Recognition, The Hague*, pages 78–82, 1992.
8. C. Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.11, No.10:1102–1106, Oct. 1989.
9. S. Negahdaripour and S. Lee. Motion recovery from image sequences using only first order optical flow information. *Intern. J. Comput. Vis.*, 9(3):163–184, 1992.
10. G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol 7:384–401, July 1985.
11. Jean-Marc Odobez. *Estimation, détection et segmentation du mouvement dans une séquence d'images: une approche robuste et markovienne.* PhD thesis, IRISA, University of Rennes I, N° 1304, December 1994.
12. P. Perez, F. Heitz, and P. Bouthemy. Multiscale minimization of global energy functions in some visual recovery problems. *CVGIP : Image Understanding*, 59(1):125–134, January 1994.
13. P.B. Chou and C.M. Brown. The theory and practice of Bayesian image modeling. *Int. Jal of Computer Vision*, Vol.4:185–210, 1990.
14. J-M. Odobez and P. Bouthemy. Detection of multiple moving objects using multiscale MRF with camera motion compensation. In *Proc.* 1st *IEEE ICIP*, volume 2, pages 257–261, Austin, Texas, November 1994.

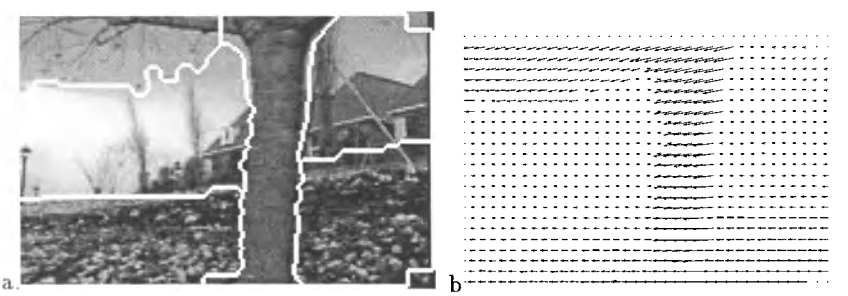


Fig. 2. "Garden sequence: segmentation map at time t_2 (a) and associated estimated model flow field (b). $m = 81; \eta = 0.3$. We obtain similar results for values of η ranging from 0.2 to 0.7.

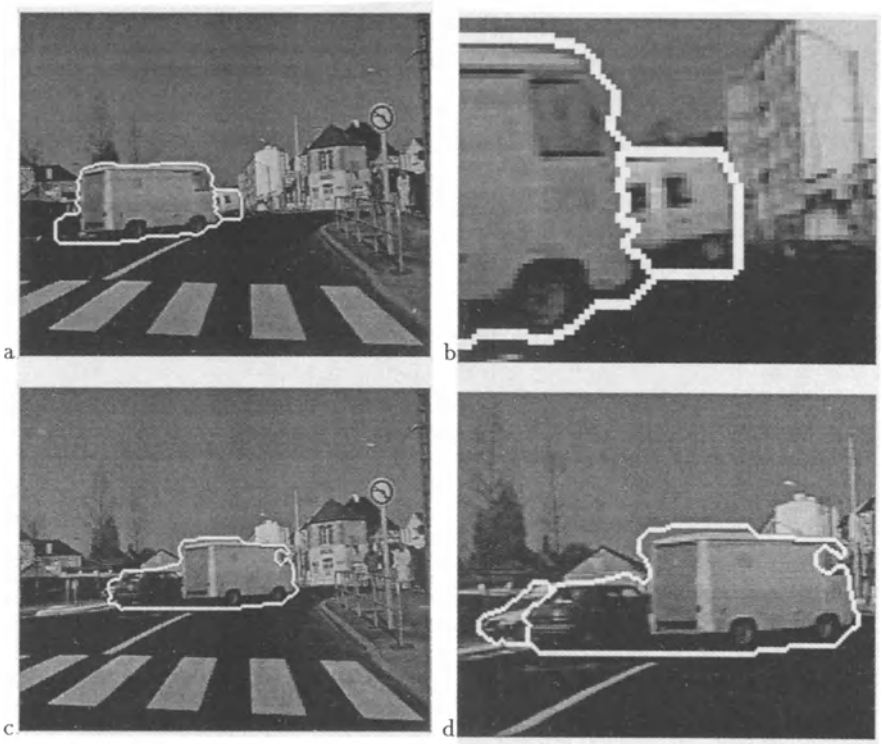


Fig. 3. "Van" sequence: a) segmentation map obtained at time t_{34}. b) detail of a). c) segmentation map obtained at time t_{53}. d) enlargement of a part of the segmentation map obtained at time t_{54}. $m = 324, \eta = 0.8$. We obtain similar results for values of η ranging from 0.5 to 1.5.

Structure from Motion

Modeling Structured Environments Using Robot Vision

Shishir Shah and J. K. Aggarwal

Computer and Vision Research Center
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712-1084, U.S.A.

Abstract. In this paper, we review the various methods for robust autonomous mobile robot navigation and scene modeling in structured environments. The techniques vary with availibility of *a priori* knowledge of the environment and the number of sensors with which the robot is equipped. The methods may be broadly classified into three categories: model based approaches, landmark based methods, and methods using information provided by the robot trajectory and its integration with sensor information for navigation and estimation of three dimensional (3D) position in the environment. Also, we describe the mobile robot *ROBOTEX*, an autonomous mobile robot constructed in our laboratory. A successful implementation for constructing computer aided design (CAD) models of a structured scene as imaged by a single wide angle lens CCD camera while navigating is considered and evaluated. Finally, we discuss another navigation system that uses a stereo pair of fish-eye lenses, and discuss the merits of such a system over some of the other implementations for navigation and modeling of the scene.

1 Introduction

The need for autonomous agents is growing as mobile robots find an increasingly large number of applications in the areas of manufacturing, hazardous materials handling, surveillance, and many others. The basic task in any such application is the perception of the environment through one or more sensors. Processing of the sensor input results in a particular representation of the environment, which can then be used for planning actions and controlling the robot. Navigation in a previously unknown environment is a quest that many vision researchers have tackled over the years. Although significant advances towards a successful system have been made, many of the existing systems for mobile robots are designed solely for path planning, obstacle avoidance, and navigation [1]. They fail to represent the environment adequately for use in different applications. Many of the existing systems require a model of the environment in order to achieve a specific goal. The vision systems discussed here are designed for autonomous navigation

* This research was supported by ARO DAAH-04-94-G-0417 and Joint Services Electronics Program Contract F49620-92-C-0027.

of a mobile robot in indoor, structured, previously unknown environments and the construction of a CAD model of the perceived environment.

The kind of environment in which a system performs significantly affects the techniques used by the robot to navigate. A wide variety of approaches and algorithms have been proposed to tackle the complex problem of indoor mobile robot navigation. The various issues that have to be addressed in the design of an autonomous agent include the basic scientific issues as well as the state-of-the-art engineering techniques [2, 3]. For successful navigation, the basic tasks of the robot can be classified as:

1. sensing the environment
2. building an environment representation
3. locating itself with respect to the environment, and
4. planning and executing efficient routes in the environment.

Because complex tasks cannot be programmed *a priori*, sensing becomes an integral part in monitoring both the environment and the robot's own internal system. The issue of navigating in an indoor structured environment has been studied by a number of other researchers [4, 5, 6, 7, 8, 9]. Different sensing modalities, such as visual sensors (monocular, binocular stereo, and trinocular stereo), infrared sensors, ultrasonic sensors, and laser range finders, have been used to aid the tasks of navigation and environment representation. The knowledge of some *a priori* information can also be used to assist the process. One of the first attemps in navigation came in 1981, when Moravec's *Cart* was used [8]. At that time, the fastest that a robot could navigate under vision control was 1 meter in 15 minutes [10]. This was certainly a big contribution at that time, but it was not fast enough for most useful applications. In general, environments modeling has been commonly achieved using sonar and infrared sensors, which requires integrating information from different sensors. The method used most often for such readings is the Bayesian certainty grid, which was introduced by Hans Moravec [11, 12]. On the other hand, most of the work in modeling using vision sensors is achieved based on detection and tracking of features relevant to the environment. The features can be tracked over time to estimate the position of the robot and the features in the environment. More qualitative approaches to model building can be achieved by recovering a graph of connections between known features in the environment.

The rest of the paper is organized as follows: Section 2 briefly presents the work that instigated the work of various researchers in the field of autonomous navigation and world map representation for indoor structured environments. Section 3 discusses the past work related to model based navigation. Section 4 reviews the techniques for navigation and world modeling using landmark based approaches. Section 5 describes previous work related to trajectory integration for navigation. We also discuss the system implemented on the mobile robot *ROBOTEX*, which is the experimental robot in our laboratory. Finally, conclusions are presented in section 6.

2 Relevant Contributions

Numerous approaches and techniques have been studied for mobile robot navigation in various environments. We will limit our review to mobile robot systems specifically designed for indoor environments. For mobile robots intended for navigation in outdoor environments, the readers are encouraged to read articles by Talluri and Aggarwal [13], Thorpe *et al.* [14, 15], Davis and Kunshner [16], Keirsey *et al.* [17], Fennema and Hanson [18], Gat *et al.* [19], and Zheng *et al.* [20].

One of the earliest attempts in indoor mobile robotics was the work of Moravec [8, 10]. Moravec used a mobile platform as a test bed for experiments in visual perception and control. The robot was remote controlled and was equipped with a single video camera. The system used a slider stereo vision system to obtain 3D spatial information for navigation. To obtain a stereo pair through the slider system, the camera was moved along a track after acquiring one image to generate the disparity field, which then could be transformed into depth information. The mobile robot, *Hilare*, designed by Giralt *et al.* [21] and Chatila and Laumond [4] used dynamic world modeling for navigation and world representation. The robot was equipped with several sensors including a video camera, a rotating sonar sensor, and a laser range finder. The robot built a world representation by incorporating any previously unknown object into a world model, as it was perceived during navigation. Heirarchical approaches using global and local model updates based on the sensor data feedback have also been suggested by Crowley [22] and Parodi [23]. Heuristic based approaches to navigation have also been explored by Chattergy [24]. In general, most algorithms rely on 3D measurements for navigation and scene modeling, but it has also been shown that similar results can be obtained by using single camera with assumptions regarding the 3D environment [25, 26, 27, 28]. Use of stereovision with line segments to recover passive 3D measurements has also been successfully used by [29, 5, 30, 31]. The use of trinocular stereo has also been explored by [32, 33, 34, 35]. Different approaches that have been studied and experimented for mobile robot navigation can be classified in three broad categories: model based approaches, landmark or reference based methods, and trajectory integration methods. We will now discuss each of these categories.

3 Model Based Approaches

Some researchers have considered robot navigation when *a priori* information about the environment is available. This may be provided in terms of a CAD model of the environment or a floor map, etc. The basic strategy then is to sense the environment using the robot sensors and to match their observations to a preloaded world map representation to arrive at an estimate of the robot pose with reduced uncertainty and to estimate a better world model.

The problem addressed by Kak et al. [36] describes the problem of matching the 3D description in CAD models to two dimensional (2D) visual images.

This is a common problem, as different forms of world map representations are used. They present PSEIKI, a system that uses evidential reasoning in a hierarchical framework for image interpretation. They describe how the system is utilized with the navigation system of the autonomous mobile robot *PETER*. The robot's position encoders maintain an approximate estimate of the robot's position and heading. To account for the wheel slippage and drift of the encoders, the visual sensor, in conjunction with the world model of the building, is used to derive a more accurate estimate. An approximate scene map is generated by using the estimate from the position encoders and the CAD model, and is then matched against the actual image viewed by the camera. Once the matches are established, the robot pose is estimated with a reduced uncertainty. The imaging geometry is such that the robot only sees fifty feet in front of the robot. PSEIKI is also able to group low level image features into high level constructs.

Once the high level features are formed, PSEIKI can match them to high level model elements. The Dempster-Shafer theory of evidence is used to accumulate evidence about the certainty of the belief values associated with each abstract. The authors also propose a computationally efficient method to overcome the combinatorial problem associated with the Dempster-Shafer formalism. A solid modeling system is used in PSEIKI to generate the expected scene maps. Complex solid objects are created. The robots pose is updated using approximate transforms as detailed by Smith and Cheeseman [37]. It is assumed that two of the three angles of rotation and one of the translation components are fixed, so that only two matches are needed to solve the position and pose problem. They also present methods for path planning in the environment.

In their later work, Kak et al. [38] present the system FINALE for Fast Indoor Navigation Allowing for Location Errors. In this approach, they use the Kalman Filter for uncertainty reduction, position estimation and updating. A geometric model of the environments is assumed, and matches between landmarks from the model and the monocular images must be determined. The system determines the approximate location of landmarks in the scene by placing bounds on where one should look for landmarks in the camera image. Given the uncertainty at any location of the robot, an expected scene model can be constructed and the uncertainties represented by the Mahanabolis distance. Each edge uncertainty regions are derived based on the vertex uncertainty regions, and the search for scene correspondents in the the model can be limited in the images and Hough space. By analyzing and processing the images only within the uncertainty regions associated with the landmarks, correspondence is quickly established. With the correspondence established and using Kalman filtering, it is possible to compute the orientation and location of the robot in the environment. The authors present real scene results and also test the accuracy of their system.

Tsubouchi and Yuta [39] discuss the techniques used in their YAMABICO robot, which uses a color camera and a map of the building in which the robot navigates. The authors present a system for combining the color image information with the map information while considering real time requirements. Their system consists of three operations. The first operation is the abstraction of

a specified image from a TV camera. The image is processed, and highly abstracted information, called the real perspective information, is generated. Next, the estimated perspective information is generated by coordinate transformation and map information, using information about the robot's position and direction. Finally, correspondence is established between the two perspectives. The authors make use of color images as they argue that they are invariant under lightness and shadow conditions. From the color images, they extract regions of similar color and fit trapezoidal regions. Trapezoids are also extracted from the map information and the two sets are used as matching primitives. Real image data is used to test their algorithms.

Fennema et al. [40] also investigated the problem of intelligent navigation for mobile robots. Given a high level command, the robot *Harvey* uses sensory input from a camera and model-based processing to carry out the assigned task. The environment in which the robot is to navigate is modeled using a solid modeling package, Geometer, which was developed jointly by University of Massachussets and GE Research and Development Center [41, 42]. The package provides explicit representation of edges, vertices, and faces, which are all parameterized by a smooth function. Surfaces are modeled as a union of these primitives. The model is contructed in a hierarchical formalism where it can be used for planning by contructing routes, monitoring by controlling the vision control to locate landmarks. The model serves as a guide for locating the robots position relative to the environment by matching primitives in the model to extracted features from the camera images.

4 Landmark Based Methods

These methods make use of landmarks in the scene to estimate the position of the robot in the environment. In general, the 3D environment can be perceived and incorporated into the model once the landmark has been identified and the respective location of the robot in the environment is triangulated with reduced uncertainty. The landmarks could be present in the environment or be placed and considered as *a priori* information for detection. The recognition of these landmarks using visual sensors is normally difficult.

Nasr and Bhanu [43] present a paradigm for navigation based on perception, reasoning, and expectation (PREACTE). They describe a knowledge-based algorithm for landmark recognition that uses an *a priori* map and the perceptual information. The perceptual information is used to create an expected model of the environment based on the approximate information of the robot's position and velocity. The expected model is matched to the image models by grouping of line and region segments and spatial reasoning. The system is adaptable to changing light and environment conditions by the provision of feedback control to low level routines. Thus, this system relies on extensive knowledge and modeled representation of the environment and the landmarks.

Anderson and Davis [44] present a similar system which uses a database of information about the environment and selects the visible landmarks by con-

trolling the camera's pose to obtain images. The landmarks are detected in the acquired images using a modified Hough transform, and their measurements are used to update the estimate of the environment and the robot's location. The system is comprised of three modules, namely the MATCHER, the FINDER, and the SELECTOR. The MATCHER identifies one or more landmarks and matches them with the information in the database. The FINDER controls the camera to detect the landmarks in the environment, and the SELECTOR picks the landmarks in the images that could be resolved to estimate the 3D information with better uncertainty.

The method presented by Bloom [45] describes the planning system for navigation based upon a grid-terrain map, a landmark database, and a landmark visibility map. The visibility map attributes the probability associated with each landmark. The planning system maintains the robot's pose based upon the land navigation system. As the position estimate worsens over time due to accumulation of error from the land navigation system, a landmark is identified from the database and a better estimate of the robot's position in the environment is triangulated by the landmark visibility map. In this paper, the landmark is attributed to color, texture, and 2D geometry. The use of these attributes in the recognition by the vision sytem is then detailed in the paper.

In the case of navigation based on a single visual camera, Sugihara [46] presents some interesting algorithms. The problem considered is that of navigation in a room given a map for navigation. He calibrates the camera with the optical axis parallel to the floor and then detects vertical edges from the scene images. The map is assumed to know the points of location for the vertical edges. He then shows that by establishing correspondence between three points in the image and three vertical lines from the map, and estimating the angles between the rays joining the image points to the lines, it is possible to uniquely determine the robot's pose.

More recently, the work presented by Bouguet and Perona [47] describe a visual navigation system using a single camera which utilizes a special detector to detect the corners of landmarks placed in the scene. By tracking these detected features, they estimate the image flow and compute the motion parameters. They use a recursive motion estimator and the 3D scene structure to recontruct the actual structure of the scene.

Most of the methods considered here for landmark-based approaches suffer from the disadvantages of:

1. assuming the availability of landmarks in the scene that the robot navigates in;
2. depending on the visibility and the ability to recognize these landmarks from the image;
3. requiring an approximate location to start with to check for landmarks; and
4. needing a database of landmarks in the area to look for in the image.

5 Trajectory Integration Methods

In this section we consider a class of techniques which estimate the robot's position by integrating its trajectory information into an evolving map or model of the environment. It is fundamental in this approach that the robot has the ability to perceive the environment, build its own representation, and be able to locate and correct its position based on the preexisting representation. These methods can prove to be limiting in some cases. We present a survey of various approaches studied using this method.

Robot sensors and the sensing strategies used are error prone and uncertain, so the model building strategy should consider these errors and account for them in a consistent way. In some of the methods, detected object features are used to form the world model. The features are detected over time with the motion of the robot and correspondence is established between the new and the old features. The robot motion, which is known through its positional sensors, is then used to predict new features in the world model. This prediction can be used to minimize the search space and establish correspondence and update the world model if needed. The type of sensing can be monocular vision, stereovision, or trinocular stereo [10, 8, 7, 32]. Moravec's *Cart* [10] uses a slider stereo system for acquiring stereo images. To locate features in a given image, the *interest operator* is used which picks regions which local maxima of a directional variance and uses this to select good features from the image. A coarse to fine correlation is used to establish correspondence between the features. Given the motion parameters of the robot, the location of features and the robot in 3D space is estimated. The world model developed by the *Cart* is the set of matched features. The uncertainty measure is scalar in this case, thus the measure is proportional to the distance of the feature from the robot location.

Matthies and Shafer [7] use an approach which is similar to Moravec's, with a spherical probability distribution of errors associated with each feature. They suggest that the 3D Gaussian distribution is a more effective way to model stereo triangulation errors. They present a method to estimate the robot position using the Gaussian error distribution. The Kalman filter is used recursively to update the locations of the features and the robot position with respect to the previously maintained world model. They assume the correspondence problem solved and show by simulation that the Gaussian error model is more accurate for stereo navigation.

Faugeras and Ayache [33, 34, 35] also propose solutions for autonomous navigation and world modeling. Their approach is mathematically more rigorous. They use a trinocular stereo system to detect relevant features in the scene, specifically the primitives used for the 3D world model are line segments. They combine coherently visual information obtained at different positions to build the 3D model of the environment. Their end result is a world map with uncertain 3D line segments attached to a reference coordinate frame and related by the uncertain motion. The measurements are further combined using the Extended Kalman Filtering technique.

Grid based methods for mapping of the environment have been used by

Moravec and Elfes [12, 48, 49]. The main idea is to represent the floor by a rectangular grid and to store information regarding the occupancy at different position on the floor as probability distributions. 3D information from the sensor provides knowledge concerning the occupied volumes in a cone in front of the robot. Occupancy information over several views is integrated into the map to make a world representation. The models accuracy improves as more readings are added, and the final map shows the regions probably occupied. This system was tested and implemented on the CMU mobile robot *Neptune*.

More recently, Lebègue and Aggarwal [50, 51, 52, 53] have presented a monocular vision system for navigation and CAD model generation of the indoor environment. They consider line segments and consider the modeling as surface patches bounded by line edges. They assume that any indoor scene and the objects within it can be represented by linear segments oriented in a limited set of directions. They segment line segments in three 3D orientations, and based on robot motion and correspondence over subsequent frames, try to recover the robot position and the positions of 3D line segments in world space. The lines are integrated over time and uncertainty is calculated using the Kalman Filtering technique. The robot updates the world representation at each step, and the final CAD model is generated when the robot has finished navigating the environment. One major drawback in this system is that, due to the limited view angle of a single wide-angle lens, the robot is not able to navigate in narrow passages or to make sharp transitions in its motion direction at hallway bends. It also cannot accurately map hollow areas in the scene. The CAD model generated by this system for a hallway is shown in Figure 1.

Shah and Aggarwal [54] present another system for autonomous mobile robot navigation and building a 3D model of an unknown, indoor structured environment. The navigation is based on binocular stereo using a pair of fish-eye lenses. No *a priori* knowledge of the environment is considered, but the inherent features of an indoor structure are exploited. Indoor structured environments can be represented by architecturally significant features and can be modeled using simple geometric primitives. The system presented assumes an obstacle-free environment and considers navigation issues in restricted spaces where the environment is narrow and has bends which require sharp manuevers of the robot. Conventional visual sensors do not provide information to make accurate measurements close to the robot, as the field of view is not large enough. They also fail to sense areas which exhibit sharp bends in the motion direction. The main characteristic of the fish-eye lens is that it has a field of view of approximately 180° in the diagonal direction. The difference in the information seen is shown by comparing two images shown in Figure 2. The image on the left is acquired through a wide angle lens and the image on the right is acquired by a fish-eye lens from the same position. It can be clearly seen that the larger field of view can provide valuable information regarding the environment which can assist in sensing objects close to the robot and in making sharp transitions in motion direction. The system is implemented on a tetherless mobile robot (*ROBOTEX*) that can navigate through narrow passages and make sharp turns in an indoor

structured environment and model the perceived scene. The system integrates all levels of image understanding, including low level image processing, so as to provide higher levels with semantically significant features.

In indoor environments, many of the objects can be represented as planar patches bounded by linear edges. The 3D orientation of the edges can be categorized in a small discrete set of possible orientations. These assumptions are incorporated as *a priori* knowledge about the environment and the algorithms developed are designed to take advantage of these observations. The linear edges in an indoor environment are considered as the boundaries of planar surfaces such as floors, ceilings, and walls. These assumptions provide a more accurate world representation while simplifying the computation. The sensors used in this case are a stereo pair of fish-eye lenses mounted on the navigating robot's platform. The lenses exhibit distortion of the imaged scene as shown in Figure 2. In order to account for the distortion and relate the camera parameters to their position on the robot, the lenses are calibrated and the distortion corrected before further processing [55]. Based on the set of possible 3D line orientations, a specialized line detector is used to extract semantically significant linear line edges [56]. The robot's egomotion is estimated from the odometry, corrected by a vision based algorithm which accurately calculates the robot's heading, roll, and pitch by using the vanishing points of the detected line segments in the scene. The stereo correspondence between the detected features in the two images is established based on an iterative hypothesis estimation and verification procedure [57]. The depth map information is utilized to navigate the robot. The calculated 3D position of the line segments provide for the best accuracy for lines close to the robot. The robot moves 1 meter at each step, before acquiring a new set of images and processing. The line segments and their 3D position are stored for modeling the navigated environment off line. Based on the uncertainty of 3D edges over several images and by considering the robot motion information, they are able to reconstruct a 3D model of the environment. Planar patches are assumed in this case to connect the line segments. In this system, the focus is on autonomous navigation for sensing indoor environments and the reconstruction of CAD models. The final reconstructed model of a hallway is shown in Figure 3.

6 Conclusion

In this paper we have illustrated various aspects of the problem of navigation and world map representation for a mobile robot. A review of methods and techniques used is provided. The techniques used vary significantly depending upon the environment in which the robot navigates, the known conditions of the environment, and the number of visual sensors with which the robot navigates. Model based methods are most applicable when it is known *a priori* the environment of the robot in the form of a world representation. The main problem to be solved in these methods is that the model and the sensor observations may be in different forms, so that techniques for matching them for recovering the robot's location and orientation have to be developed. Landmark based methods are suitable for

navigating in environments where landmarks can be identified. This usually requires the robot to have a database of landmarks occurring in the environment and an approximate location to start the database search. It is also possible to use these methods if the landmark can be identified by pre-specified features in the environment. Methods using trajectory integration usually require the robot to address the problem of environment perception and modeling. The different methods studied to model the environment and form a map of it for navigational tasks are also reviewed in this paper. We have also discussed the trajectory integration approach for autonomous navigation that has been implemented on a mobile robot in our laboratory. The merits of our system are compared to some of the other existing methods and real scene results are presented.

References

1. T. Dean and R. P. Bonasso, "1992 AAAI robot exhibition and competition," *AI Magazine*, vol. 14, pp. 35–48, Spring 1993.
2. I. J. Cox and G. T. Wilfong, *Autonomous Robot Vehicles.* New York: Springer–Verlag, 1990.
3. S. S. Iyengar and A. Elfes, *Autonomous Mobile Robots.* Los Alamitos, CA: IEEE Computer Society Press, 1991.
4. R. Chatila and J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, (St.Louis), pp. 138–145, March 1985.
5. J. L. Crowley, "Dynamic world modeling for an intelligent mobile robot using a rotating ultra-sonic ranging sensor," in *Proc. IEEE Int. Conf. Robotics and Automation*, (St.Louis), pp. 128–135, 1985.
6. O. D. Faugeras and N. Ayache, "Building, registering and fusing noisy visual maps," in *Proc. First Int. Conf. Computer Vision*, (London, UK), pp. 73–82, June 1987.
7. L. Matthies and S. A. Shafer, "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. RA-3, pp. 234–248, June 1987.
8. H. P. Moravec, *Robot Rover Visual Navigation.* Ann Arbor, MI: UMI Research Press, 1981.
9. T. Tsubouchi and S. Yuta, "Map assisted vision system of mobile robots for reckoning in a building environment," in *Proc. IEEE Int. Conf. Robotics and Automation*, (Raleigh), pp. 1978–1984, March 1987.
10. H. P. Moravec, "The stanford cart and cmu rover," *Proceedings of IEEE*, vol. 71, no. 7, pp. 872–884, 1983.
11. H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, pp. 61–74, Summer 1988.
12. H. P. Moravec and D. W. Cho, "A bayesian method for certain grids," in *Proc. of AAAI Spring Symposium Series on Mobile Robot Navigation*, (Satnford), April 1989.
13. R. Talluri and J. K. Aggarwal, "Autonomous navigation in cluttered outdoor environments using goemetric visibility constraints," *Proc. of the Intl. Conf. on Intelligent Autonomous Systems*, vol. IAS-3, February 1993.

14. C. Thorpe, S. Shafer, and T. Kanade, "Vision and navigation for the Carnegie Mellon Navlab," in *Proc. Image Understanding Workshop, DARPA*, (Palo Alto), pp. 143–152, February 1987.
15. C. Thorpe and T. Kanade, "Carnegie Mellon Navlab vision," in *Proc. Image Understanding Workshop, DARPA*, (Palo Alto), pp. 273–282, May 1989.
16. L. Davis and T. R. Kunsher, "Vision-based navigation: A status report," *Proceedings of Image Understanding Workshop*, pp. 153–169, February 1987.
17. D. M. Keirsey, D. W. Payton, and J. K. Rosenblatt, "Autonomous navigation in cross country terrain," in *Proc. Image Understanding Workshop, DARPA*, (Cambridge), pp. 411–416, April 1988.
18. C. Fennema and A. R. Hanson, "Experiments in autonomous navigation," in *Proc. Tenth Int. Conf. on Pattern Recognition*, (Atlantic City, New Jersey), pp. 24–31, June 1990.
19. E. Gat, M. G. Slack, R. J. Firby, and D. P. Miller, "Planning for execution monitoring on a planetory rover," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 20–25, 1990.
20. J. Y. Zheng, M. Barth, and S. Tsuji, "Autonomous landmark selection for route recognition by a mobile robot," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 2004–2009, 1991.
21. G. Giralt, R. Sobek, and R. Chatila, "A multi-level planning and navigation system for a mobile robot; a first approach to hilare," in *Proc. Sixth International Joint Conference on Artificial Intelligence*, pp. 335–337, 1979.
22. J. Crowley, "Navigation of an intelligent mobile robot," *IEEE Journal of Robotics and Automation*, vol. 31, no. 1, 1985.
23. A. Parodi, "Multi-goal real-time global path planning for an autonomous land vehicle using a high speed graph search processortion," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (St. Louis, Missouri), pp. 161–167, 1985.
24. A. Chattergy, "Some heuristics for the navigation of a robot," *Robotics Research*, vol. 4, pp. 59–66, Spring 1985.
25. P. Belluta, G. Collini, A. Verri, and V. Torre, "3D visual information from vanishing points," in *Proc. IEEE Workshop on Interpretation of 3D Scenes*, (Austin, Texas), pp. 41–49, November 1989.
26. P. Belluta, G. Collini, A. Verri, and V. Torre, "Navigation by tracking vanishing points," in *Working Notes of AAAI Robot Navigation Symposium*, (Stanford University), pp. 6–10, March 1989.
27. P. Olivieri, M. Gatti, M. Straforini, and V. Torre, "A method for the 3D reconstruction of indoor scenes from monocular images," in *Proc. Second European Conf. on Computer Vision*, (Santa Margherita Ligure, Italy), pp. 696–700, Springer-Verlag, May 1992.
28. M. Straforini, C. Coelho, M. Campani, and V. Torre, "The recovery and understanding of a line drawing from indoor scenes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 298–303, February 1992.
29. J. L. Crowley, "World modeling and position estimation for a mobile robot using ultra-sonic ranging," in *Proc. of the IEEE Int. Conf. Robotics and Automation*, (Scotsdale, Arizona), pp. 674–680, May 1989.
30. H. Ishiguro, M. Yamamoto, and S. Tsuji, "Omni-directional stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 257–262, February 1992.
31. D. Kim and R. Nevatia, "Indoor navigation without a specific map," in *Proc. Int. Conf. Intelligent Autonomous Systems*, (Pittsburgh, PA), pp. 268–277, February 1993.

32. A. R. de Saint Vincent, "A 3D perception system for the mobile robot hilare," in *Proc. of the IEEE Int. Conf. Robotics and Automation*, (San Francisco), pp. 1105 – 1111, 1986.
33. N. Ayache and O. D. Faugeras, "Building a consistent 3D representation of a mobile robot environment by combining multiple stereo views," in *Proc. 10th International Joint Conference on Artificial Intelligence, IJCAI '87*, pp. 808–810, 1987.
34. N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. on Robotics and Automation*, vol. 5, pp. 804–819, December 1989.
35. O. D. Faugeras, N. Ayache, and B. Faverjon, "Building visual maps by combining noisy stereo measurements," in *Proc. of the IEEE Int. Conf. Robotics and Automation*, (San Francisco), pp. 1433–1438, 1986.
36. A. Kak, K. Andress, and C. Lopez-Abadia, "Mobile robot self-location with the PSEIKI system," in *Working Notes, AAAI Spring Symposium on Robot Navigation*, (Cambridge), pp. 33–37, March 1989.
37. R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. Journal of Rob. Res.*, vol. 5, no. 4, pp. 56–58, 1987.
38. A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 56, pp. 271–329, November 1992.
39. T. Tsubouchi and S. Yuta, "Map assisted vision system of mobile robots for reckoning in a building environment," in *Proc. IEEE Int. Conf. Robotics and Automation*, (Raleigh, North Carolina), pp. 1978–1984, March 1987.
40. C. Fennema, A. R. Hanson, and E. Riseman, "Towards autonomous mobile robot navigation," in *Proc. of the SPIE – International Society for Photographic and Industrial Engineering*, pp. 219–231, 1990.
41. C. Connolly, "Geometer: Solid modelling and algebraic manipulation," in *Technical Report*, (University of Massachusetts, Amherst), 1989.
42. C. Connolly and R. Weiss, "Geometer: Solid modelling and algebraic manipulation," in *Proc. of the Image Understanding Workshop, DARPA*, (Palo Alto, CA), 1989.
43. H. Nasr and B. Bhanu, "Landmark recognition for autonomous mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1218–1223, 1988.
44. F. P. Angerson and L. S. Davis, "Visual position determination for autonomous vehicle navigation," in *Center for Automation Research, Tech Report, CAR-TR-100*, November 1984.
45. B. C. Bloom, "Use of landmarks for mobile robot navigation," in *Proc. SPIE Vol. 579 Intelligent Robots and Computer Vision*, pp. 351–355, 1985.
46. K. Sugihara, "Some location problems for robot navigation using a single camera," *Computer Vision, Graphics and Image Processing*, vol. 42, pp. 112–129, April 1988.
47. J.-Y. Bouguet and P. Perona, "Visual navigation using a single camera," in *Proc. International Conference on Computer Vision*, pp. 645–652, June 1995.
48. L. Matthies and A. Elfes, "Integration of sonar and stereo range data using a grid based representation," in *Proc. of the IEEE Int. Conf. Robotics and Automation*, (Philadelphia), pp. 727–733, April 1988.
49. A. Elfes, "Sonar based real-world mapping and navigation," *IEEE Jour. of Robotics and Automation*, vol. RA–3, pp. 249–265, June 1987.
50. X. Lebègue and J. K. Aggarwal, "A mobile robot for visual measurements in architectural applications," in *Proc. IAPR Workshop on Machine Vision Applications*, (Tokyo, Japan), pp. 195–198, December 1992.

51. X. Lebègue and J. K. Aggarwal, "Robotex: An autonomous mobile robot for precise surveying," in *Proc. Int. Conf. Intelligent Autonomous Systems*, (Pittsburgh, PA), pp. 460–469, February 1993.
52. X. Lebègue and J. K. Aggarwal, "Automatic creation of architectural CAD models," in *Proc. 2nd CAD-Based Vision Workshop*, (Seven Springs, PA), February 1994.
53. X. Lebègue and J. K. Aggarwal, "Architectural CAD modeling using a mobile robot." Submitted to *IEEE Transactions on Robotics and Automation.*
54. S. Shah and J. K. Aggarwal, "Autonomous mobile robot navigation using fish-eye lenses." Submitted to *Proc. Third International Computer Science Conference*, December 1995.
55. S. Shah and J. K. Aggarwal, "A simple calibration procedure for fish-eye (high distortion) lens camera," in *Proc. of Int. Conf. on Robotics and Automation*, (San Diego, California), pp. 3422–3427, 1994.
56. X. Lebègue and J. K. Aggarwal, "Detecting 3-D parallel lines for perceptual organization," in *Proc. Second European Conf. on Computer Vision*, (Santa Margherita Ligure, Italy), pp. 720–724, Springer-Verlag, May 1992.
57. S. Shah and J. K. Aggarwal, "Depth estimation using stereo fish-eye lenses," in *Proc. of Int. Conf. on Image Processing*, (Austin, Texas), pp. 740–744, 1995.

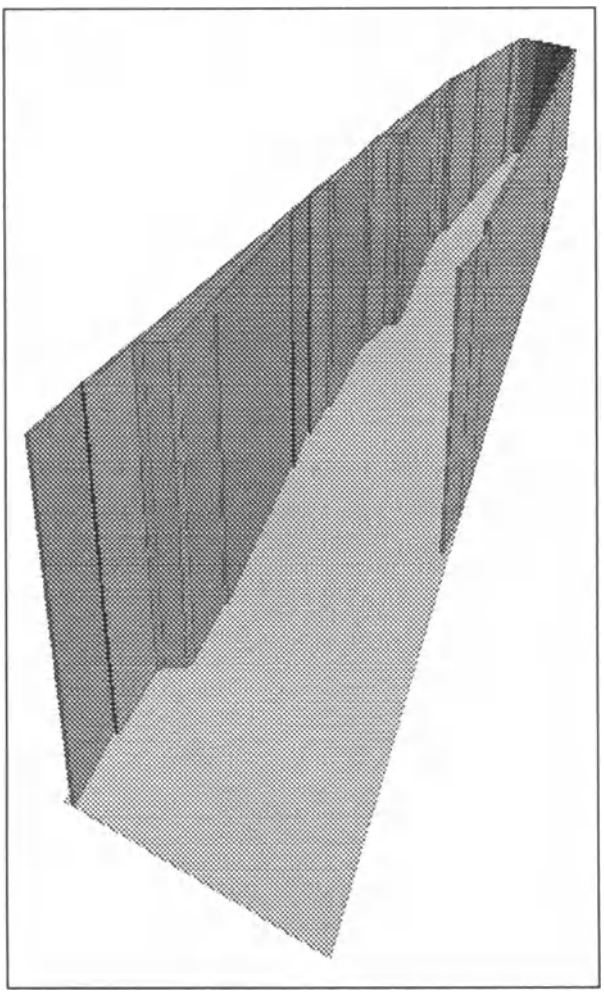
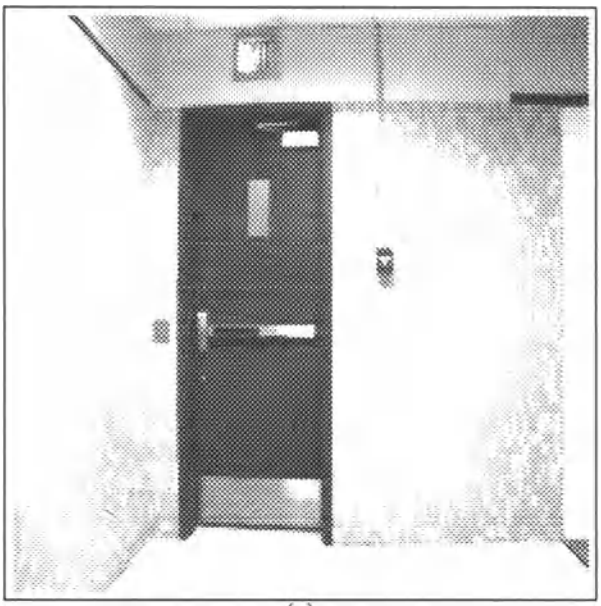


Fig. 1. CAD model of hallway by visual navigation using a single wide angle lens.



(a)

(b)

Fig. 2. (a) Image taken by a wide-angle lens at corridor end. (b) Fish-eye lens image taken from the same position contains more information about the corridor.

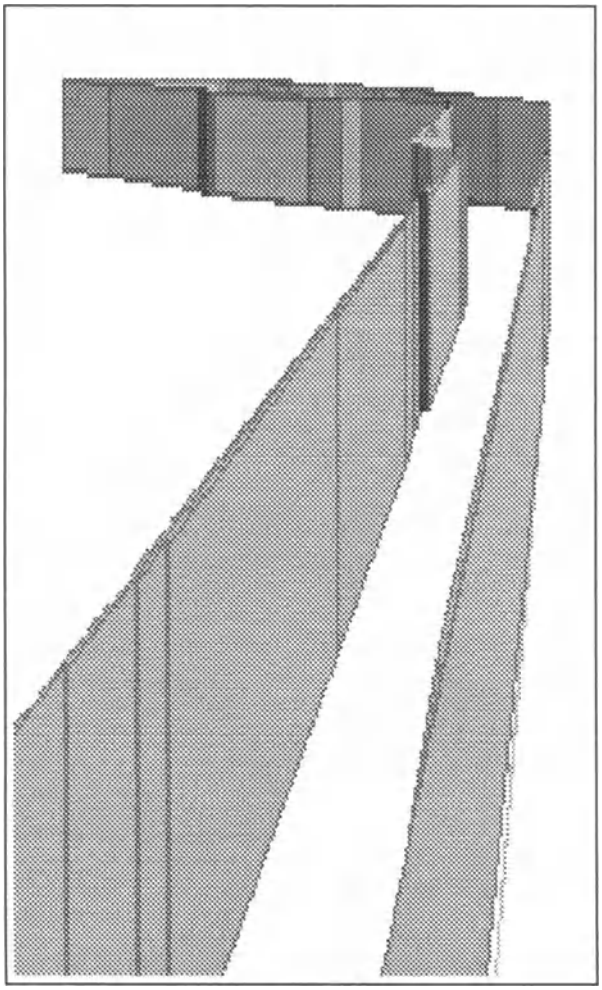


Fig. 3. CAD model of hallway by visual navigation using stereo fish-eye lenses.

Design Trade-Offs for Real-Time Block-Matching Motion Estimation Algorithms

Zhong L. He and Ming L. Liou
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

Abstract: In this paper, VLSI design trade-offs for real-time block-matching motion estimation (BMME) algorithms are discussed. Traditionally, design trade-offs are considered only for two design parameters, i.e, processing speed and silicon area. However, this is not sufficient for real-time BMME algorithms. In addition to the traditional ones, we introduce two new design parameters, i.e, I/O bandwidth and algorithmic performance for consideration. Various design trade-offs among these four parameters are presented in this paper.

1 Introduction

Motion estimation technique is widely used in both computer vision and video compression. Numerous motion estimation algorithms have been proposed in the past. In computer vision applications, motion estimation is used for dynamic scene analysis since it provides important information that is needed in target tracking, vehicle navigation, and recognition of 3-D objects[1]. Most motion estimation algorithms in computer vision are designed for software execution. They possess very complex data structures thus are not amenable for cost effective VLSI implementation. In video compression, motion estimation is employed for removing temporal redundancy between frames. Real time execution is often required in most applications. Motion estimation for video compression also requires a great deal of computation. Consequently, it is not suitable for software execution. Among various motion estimation methods, block-matching motion estimation (BMME) technique is the most widely used method and is suitable for cost effective VLSI implementation. In BMME, each frame of the video sequence is partitioned into small non-overlap blocks. Block-matching is performed by comparing a block in a current frame with blocks in an already processed frame within a search window to find the best match by a certain criterion. BMME frequently fails to produce good results in computer vision applications because it relies on data only and does not attempt to explicitly model a motion field[2].

For real-time video compression applications, a BMME algorithm is normally implemented with dedicated VLSI circuits. Due to the very complex nature of VLSI design space, there exist many design trade-offs. Proper consideration of the trade-offs can lead to an optimal VLSI architecture design for a particular application. In this paper, we will discuss the VLSI design trade-offs for BMME algorithms at the architecture level. In addition to the processing speed and

silicon area which we normally use as measures for VLSI design, we will introduce two new measures, i.e., I/O bandwidth and performance, for consideration. Some newly proposed architectures will be analyzed for comparison.

2 The BMME technique

Fig. 1 describes the block-matching motion estimation algorithm. Suppose the block size is $N \times N$ and the maximum displacement is $\pm p$ pixels. For the ease of VLSI implementation, we choose the mean absolute difference (MAD) of pixels between the two matching blocks as the matching criterion. The number of blocks to be matched in the search window is $(2p+1)^2$. The full search approach is to compare the current block with all the blocks in the search window to determine the best match with minimum MAD. The relative position of this block from the corresponding current block is called *motion vector*. Because all the blocks in the search window are compared, the full search algorithm always yields the optimal solution. However, it suffers from high computation complexity. Fast algorithms compare the current block with some specific blocks in the search window thus can speed up the computation. However, they can be easily trapped in a local minimum thus deliver inferior performance.

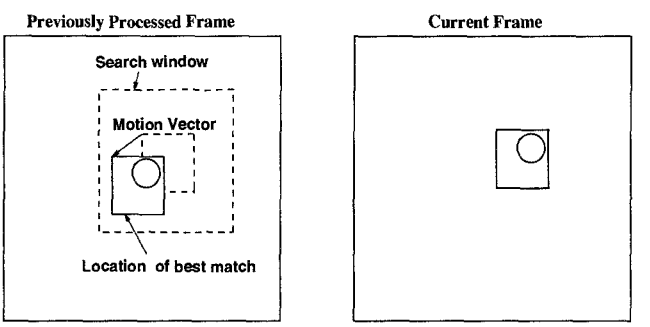


Fig. 1. Description of block-matching motion estimation algorithm

3 Design parameters for real-time BMME algorithms

For VLSI implementation, two important design parameters are normally considered. One is processing speed and the other is silicon area. However, for the real-time VLSI implementation of BMME algorithms, using only these two parameters as design metrics is not sufficient. In addition to these two parameters, we propose two new parameters, i.e., I/O bandwidth and algorithmic performance, for consideration. These design parameters are discussed below.

3.1 Processing speed

For real-time applications, processing speed or throughput is the most important parameter for VLSI implementation of BMME algorithms. Computation required for a particular BMME algorithm has to be completed at the rate not slower than the pixel rate. Given a VLSI technology, processing speed of a particular architecture or VLSI implementation can be appropriately measured by the number of clock cycles needed for one current block matching, the number of absolute difference operations completed per second, or the number of current block matching per second. Since the current technology can't handle block matchings serially in real-time, various forms of parallel architecture are needed. This would increase the size of silicon area. Too much of parallelism may exceed the physical limitations of a chip. Processing speed also depends on the type of application and the required algorithmic performance. It differs a great deal between HDTV and videophone.

In the following, we consider the processing speed for the full-search BMME algorithm. Suppose the block size is 16×16 and the search range is $p = -8, 7$. The required throughput for H.261 applications (352×288, 30 frames/sec) is $\frac{352 \times 288}{16 \times 16} \times (7+8+1)^2 \times (16 \times 16) \times 30 = 778M$ absolute difference operations per second. (In fact, throughput depends only on the picture size, search window size, and frame rate, not the block size). Table 1 shows the comparison of two VLSI implementations (using 1-D systolic array) for H.261 applications[5][6], where the throughput is measured by the number of absolute difference operations per second. The throughput in each implementation is achieved by using parallel PEs (Processing Elements) operated at almost full efficiency. Clearly, a single chip implementation using [6] is good enough for H.261 applications while two chips are required for [5].

Table 1. Comparison of chips for the implementations of full search motion estimation

Imple-mentation	No.of PE	Core area	Clock rate	Throughput ops/sec
Imple. in [5]	16	$51(mm)^2$	25MHZ	400M
Imple. in [6]	32	$79(mm)^2$	50MHZ	960M

For the MPEG-2/HDTV applications, the two implementations in Table 1 are not fast enough. Therefore, a much higher throughput architecture with large search window is necessary. Several VLSI architectures for the full search algorithm with high throughput have been proposed in recent years[7][8][9]. These architectures are based on 2-D systolic array and the number of PEs in these architectures equals to the number of pixels in a matching block. Table 2 shows the main characteristics of the architecture proposed in [8] with the search window $\pm 32H \times \pm 16V$. The throughput of this architecture is proportional to the

number of pipelined stages used in each PE. Each PE in this architecture works in 3 or 4 pipelined stages and the proposed architecture can meet the requirements of MPEG-2/HDTV applications. However, compared to the 1-D systolic array, the number of PEs is increased and also is the silicon area.

Table 2. Main characteristics of a 2-D systolic array architecture[8] (pipelined clock rate 100MHz)

Pipelined stages	No.of PE	No. of trans.	Data access cycle	Throughput ops/sec
3	256	688k	30ns	25600M
4	256	823k	40ns	25600M

3.2 Silicon area

The size of silicon area depends on technology as well as on the BMME algorithm and VLSI design. It can't be determined precisely before a chip is laid out with a specific technology or design rule. At the architecture level, the size of silicon area may be estimated by the total number of gates required for the entire operations including interconnections such as wires, buses, and multiplexers. Cost of VLSI chips depends directly on the size of silicon area.

Given a specific application and required performance, minimization of silicon area largely depends on the architecture design taking into the consideration of I/O bandwidth limitation. Design regularity, efficient utilization of circuits, and the need of on-chip memory become important issues for consideration. For example, we consider three VLSI architectures[10][11][12] for the three-step search (TSS) algorithm for H.261 applications. Referring to Fig. 2, the architecture in 2(a) is based on 1-D systolic array and programmable delay unit (PDU). This architecture can achieve cost effective VLSI implementation with high throughput, reasonable I/O bandwidth and regular interconnections. The architecture in Fig. 2(b) uses only 9 PEs for parallel processing with 9 on-chip memory modules. The architecture shown in Fig. 2(c) is a tree structure with different levels. The top level is for absolute difference operations and the others are for addition and accumulation operations. The operations for computing MADs are pipelined through different levels.

Table 3 shows the complexity comparison among these three architectures. It should be noted that the number of PEs in [12] is not actual but equivalent. This way we can make a fair comparison with the other architectures. It should also be noted that the gate count in Table 3 includes both the PEs and on-chip memories but not interconnections.

Compared to the architecture in [11], the gate count in [10] is not significantly reduced. However, the interconnections in [10] are simpler, and the overall silicon

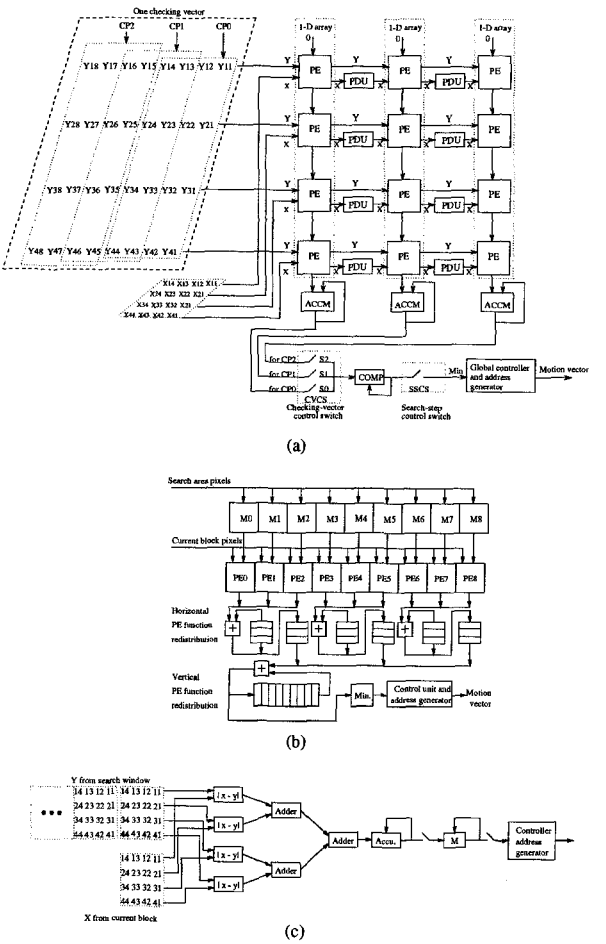


Fig. 2. Architectures for the TSS algorithm and (a) 1-D systolic array based architecture[10], (b) 9PE architecture with on-chip memory[11], and (c) tree architecture[12]

area can be saved. Among these architectures, the gate count in [12] is the smallest and the throughput is moderate. However, the required I/O bandwidth is very high which will heavily limit the processing speed.

3.3 I/O bandwidth

The I/O bandwidth of a VLSI implementation for BMME algorithms is determined largely by the pixel rate of search window which resides outside of the BMME chip. Certain pixels need to be input repeatedly due to a specific architecture design, thus increase the I/O bandwidth. I/O bandwidth may be reduced at the expense of increased size of silicon area by allowing on-chip memory for the storage of pixels from the search window. Referring to Table 3, the architec-

Table 3. Complexity comparison of architectures for the TSS algorithm (The throughput and the I/O rate are measured by the number of clock cycles and the number of input pixels from search window per current block-matching respectively)

Architectures	Archi. in [10]	Archi. in [11]	Archi. in [12] $\frac{1}{16}$-cut tree
No. of PEs	48	9	25(equivalent)
On-chip mem. size (No. of gates)	8k	22k	0
Gate count	32.9k	36.6k	9.54k
Throughput	230	794	456
I/O rate	2976	1024	6912

ture in [12] does not use on-chip memory to store the pixels for reuse. Each block in the search window is independently processed and no data dependency is exploited. Therefore, the required I/O bandwidth is high and the chip has to use expensive off-chip memory to meet the real-time requirement. The architecture in [11] uses 60% equivalent gates for the on-chip memory design to reduce the I/O bandwidth. However, this on-chip memory occupies large silicon area and more control circuit is employed for the exchange of the partial summations.

The I/O bandwidth can be significantly reduced for certain search algorithms with specific data flow and architecture design. Considering the architecture in [10], it reduces the I/O bandwidth by exploiting the data dependency between the blocks in search window. By using appropriate number of PEs (3N PEs for $N \times N$ block size) and on-chip memory (DPUs) to temporarily store and broadcast input pixels, the I/O bandwidth can be significantly reduced.

3.4 Performance

As mentioned earlier, there are two categories for BMME algorithms. One is full-search and the other is fast-search. Since the full-search algorithm requires huge computation, fast-search algorithms have been developed as alternate solutions for certain applications. However, there are two drawbacks for the existing fast-search algorithms. One is that the solution may not be optimal and the other is that the data flow is irregular.

Clearly, the four design parameters mentioned above are related to each other. For a single chip VLSI implementation, these four design parameters should be considered together. In the next section, we will introduce several trade-off measures for these four parameters.

4 VLSI design trade-offs

Traditionally, VLSI design trade-offs are considered for only two parameters, i.e. processing speed and silicon area. For real-time BMME applications, we need to

consider two new parameters, i.e., I/O bandwidth and algorithmic performance. Trade-offs among these parameters will be discussed in this section.

4.1 Area-Time trade-off

As mentioned earlier, silicon area and processing time should be as small as possible. However, they are often in conflicting situations. In order to reduce chip area, processing time is usually increased and vice versa. Therefore, the Area-Time (A-T) product $A^{\alpha} \times B^{\beta}$, where $\alpha > 0$, $\beta > 0$ is a reasonable measure for the design trade-off. Clearly, the A-T trade-off metric is only suitable for the VLSI architectures or implementations with the same I/O bandwidth and the same algorithmic performance.

For example, we consider the two architectures for the TSS algorithm shown in Fig. 3 (assuming that, for simplicity, the block size is 4×4) and also the one shown in Fig. 2(c). We assume that only the external memories are used and thus these architectures have the same memory bandwidth because they have the same data flow of the TSS algorithm and no data dependency is used. These three architectures in Fig. 3(a), 3(b) and 2(c) are called 1-D systolic array[7], 9PE-architecture in [11] with no on-chip memory, and tree structure[12] respectively. Table 4 shows the A-T products with different α and β values normalized with respect to the architecture of 1-D systolic array when the block size is 16×16. For ease of discussion, the silicon area is simply measured by the total number of gate count. Clearly, these three architectures in terms of $A \times T$ metric are not too much different. The 1-D systolic array and tree architecture are better than the 9-PE architecture in terms of $A \times T^2$ metric.

Table 4. The A-T trade-off comparison for the TSS algorithm (A in number of gates, T in number of clock cycles)

Architecture	1-D array	Tree archi.	9-PE archi.
A	$8k$	$9.54k$	4.5k
T	477	456	768
AT	1	1.13	0.91
AT^2	1	1.07	1.46

4.2 Area-Time-Bandwidth trade-off

For the architectures with difference I/O bandwidth, the A-T trade-off becomes ineffective. In this case, we propose the Area-Time-Bandwidth (A-T-B) trade-off $A^{\alpha} \times T^{\beta} \times B^{\gamma}$, where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$, as a design metric.

Table 5 lists the comparison results for the three VLSI architectures of the TSS algorithm listed in Table 3. Obviously, the architecture in [10] is the best

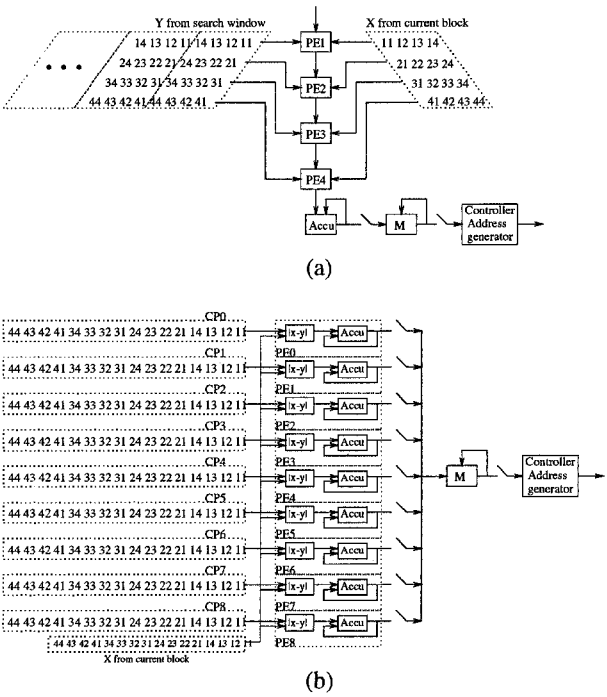


Fig. 3. Architectures with the same I/O bandwidth for the TSS algorithm with block size 4×4 and (a) 1-D systolic array, and (b) 9PE-architecture with no on-chip memory

choice if $\alpha = \gamma = 1$ and $\beta \geq 1$. The architectures in [11] and [12] is the best design if we emphasize I/O bandwidth and silicon area respectively.

4.3 Area-Time-Bandwidth-Performance trade-off

The A-T-B product metric is reasonable for the architectures with the same algorithm. However, for a specific application such as H.261, MPEG, or HDTV,

Table 5. The A-T-B trade-off comparison for the architectures of the TSS algorithm normalized with respect to the architecture in [10]

Architectures	Archi. in [10]	Archi. in [11]	Archi. in [12] $\frac{1}{16}$-cut tree
ATB	1	1.31	1.31
ATB^2	1	0.45	3.05
AT^2B	1	4.44	2.56
A^2TB	1	1.47	0.38

many architectures with different algorithmic performance can be applied. For example, for the H.261 applications, the architectures of TSS, NTSS (new three-step search[3]), or full-search algorithms are all suitable. In this case, the A-T-B product metric is not sufficient and a new trade-off metric should be used.

Unfortunately, there is no simple and accurate way to measure the algorithmic performance. Numerically, we can use MAD or some other means to indicate the algorithmic performance. However, it is video source dependent. In many cases, we have to rely on subjective judgement. Generally speaking, performance is very important for high quality video applications, such as MPEG-2 and HDTV. For videophone and video conferencing applications, performance can be relaxed and fast search techniques can be applied in exchange of a smaller A-T-B product.

5 Conclusion

In this paper, we have introduced four important parameters and their trade-offs for VLSI design of real-time block-matching motion estimation algorithms. In addition to the traditional two design parameters, i.e., processing speed and silicon area, we have proposed two new design parameters, I/O bandwidth and algorithmic performance. Based on the comparison results shown in this paper, we believe that the proposed design parameters and trade-off metrics are very effective to measure various VLSI architectures or implementations for a specific video application.

References

1. Sun Xuening and Minas E. Spetsakis, "A comparison of weighted LS methods with LS methods in 3-D motion estimation from stereo image sequences", *Proceedings 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.200-205, 1993.
2. Janusz Konrad and Eric Dubois, "Bayesian estimation of motion vector fields", *IEEE transactions on Pattern analysis and Machine Intelligence*, vol. 14, No. 9, Sept. 1992.
3. R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for fast block motion estimation ", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 4, No. 4, pp.438-442, Aug. 1994.
4. T. Koga, K. Iinuma, A, Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing", *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov. 29, Dec. 1981, pp. G5.3.1-5.3.5.
5. K. M. Yang, M. T. Sun and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm", *IEEE Transactions on Circuits and Systems*, vol. 36, No. 10, pp.1317-1325, Oct. 1989.
6. P. Ruetz, P. Tong, D. Bailey, D. Luthi and P. Ang, "A high performance full-motion video compression chip set", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 2, pp.111-122, June 1992.

7. T. Komarek and P. Pirsch, "Array architecture for block matching Algorithms", IEEE Transaction on Circuits and Systems, Vol. 36, No. 10, pp.1301-1308, 1989.
8. Z. He and M. L. Liou, "A new array architecture for motion estimation", *Proceedings of IEEE Workshop on Visual Signal Processing and Communications*, pp.148-151, Sept. 1994.
9. L. D. Vos and M. Stegherr, "Parameterizable VLSI architecture for the full-search block-matching algorithm", *IEEE Trans. on Circuits and System*, Vol. 36, No. 10, pp. 1309-1316. 1989.
10. Zhong L. He and Ming L. Liou, P. C. Chan, and R. Li, "An efficient architecture for a new three-step search algorithm", to appear in *Proceedings of 38th Midwest Symposium on Circuit and systems*, 1995.
11. H. M. Jong, L. G. Chen, T. D. Chiueh,"Parallel architectures for 3-step hierarchical search block-matching algorithm", *IEEE Trans. on Circuits and System for Video Technology*, vol. 4, No. 4, pp.407-416, August 1994.
12. Y. S. Jehng, L. G. Chen and T. D. Chiueh, "An efficient and simple VLSI tree architecture for motion estimation algorithms", *IEEE Trans. on Signal Processing*, Vol. 41, No. 2, pp.889-900, Feb., 1993.

Pixel Matching and Motion Segmentation in Image Sequences

Narendra Ahuja and Ram Charan

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign, Urbana, Illinois, U.S.A. 61801
e-mail: ahuja@stereo.ai.uiuc.edu, ramm@stereo.ai.uiuc.edu

Abstract. This paper presents a coarse-to-fine algorithm to obtain pixel trajectories in a long image sequence and to segment it into subsets corresponding to distinctly moving objects. Much of the previous related work has addressed the computation of optical flow over two frames or sparse feature trajectories in sequences. The features used are often small in number and restrictive assumptions are made about them such as the visibility of features in all the frames. The algorithm described here uses a coarse scale point feature detector to form a 3-D dot pattern in the spatio-temporal space. The trajectories are extracted as 3-D curves formed by the points using perceptual grouping. Increasingly dense correspondences are obtained iteratively from the sparse feature trajectories. At the finest level, which is the focus of this paper, all pixels are matched and the finest boundaries of the moving objects are obtained.

Keywords: Motion Segmentation, Perceptual Grouping, Pixel Matching, Triangulation, Feature Matching, Optical Flow.

1 Introduction

This paper describes a component of our work aimed at interpretation of image sequences. Given an image sequence containing an arbitrary number of rigid objects in motion, the objectives of the overall work are to identify feature points in the scene, obtain spatially dense trajectories of those points, segment moving objects, compute image flow at each pixel, and derive a qualitative description of the scene structure and dynamics from the image sequence. Such qualitative interpretation of the image sequence is useful for a variety of applications such as traffic scene analysis, biological image analysis and aerial image understanding. The focus of this paper is on the detection of pixel flow trajectories. Next section reviews some related previous work. Section 3 summarizes the steps of coarse-to-fine detection of sparse feature trajectories. Section 4 then gives the details of the algorithm for finding pixel flow trajectories which is the objective of this paper. Section 5 presents experimental results and Section 6 presents concluding remarks.

2 Previous Work

Much previous work has addressed feature correspondences between two frames which are used for 3D motion and structure estimation. The work on optical flow has been concerned with the detection of pixel correspondences across two frames closely separated in time. Both of these areas have seen a significant amount of activity and we will not attempt to provide the long list of references here. For motion analysis from long image sequences, several researchers have addressed the problem of feature correspondence in the past. Sethi and Jain[1] formulate this problem as an optimization problem and propose an iterative algorithm which they call the Greedy Exchange algorithm. Sethi et al.[2] propose a relaxation algorithm for feature point matching where the formation of smooth trajectories over space and time is favored. This method requires the correct initial correspondence and was used on very few feature points. Rangarajan and Shah [3] have proposed a noniterative polynomial time approximation algorithm by minimizing a proximal uniformity cost function. Cheng and Aggarwal[4] propose a two stage hybrid approach to the trajectory finding problem. The first stage extends the trajectories and the second one attempts to correct any errors. Debrunner and Ahuja[5] uses a two stage method to finding trajectories. The first step computes short feature paths of constant velocity. The second step deals with joining the feature paths into trajectories.

This paper exploits known trajectories of features for finding pixel trajectories. In the next section, we first briefly present our approach to coarse-to-fine feature trajectory detection which is based on perceptual grouping.

3 Feature point matching and segmentation

This section summaries the first two steps of the algorithm to analyze a single batch of frames.

3.1 Sparse Correspondences

The goal of the first step is to find correspondences of sparsely located feature points and segment the moving objects. We have no knowledge of number of moving objects or their motions in the scene. The only assumptions made here are that they are rigid objects and are moving smoothly. A perceptual grouping technique is used to achieve this goal. Feature points are detected in each image. The images in a batch are stacked to form a 3D dot pattern. The 3D Voronoi tessellation defined by the points is constructed. The batch size must be chosen carefully. If the number of frames in the batch is small, then the Voronoi tessellation may not represent the 3D structure well due to lack of data along time axis. A large batch size, on the other hand, may contain frames in which a moving object enters or exits the visual field which results in trajectories that last for only part of the batch. This makes their detection more difficult. Further, it significantly adds to the computational load. Since in the algorithm used in this

work, the batch analysis is used to estimate final correspondences for a central pair of frames, it is sufficient to ensure that the Voronoi structure associated with the dots in several central frames are correct, i.e., are not affected by lack of image frames.

The result of this step is a curve segment joining the feature points in all frames. In general, some feature points are missed and some new feature points appear from frame to frame. Therefore, the length of a curve segment or trajectory varies. The feature point locations along a trajectory in the two central frames are considered as correspondences. Such correspondences may be used by any of a number of motion and structure algorithms [6, 7] that require point feature correspondences.

Once the feature point correspondences are known, the feature points in each frame are segmented into different moving objects based on similarity of motion. Local adjacency among points is made explicit through the Delaunay triangulation when a Delaunay edge connects a point with its Voronoi neighbors. Segmentation is then achieved by identifying Delaunay edges connecting points belonging to different objects as well as those inside a single object. In general two independently moving objects differ in the magnitude and direction of their 3D motion. However 2D direction alone is a strong basis to discern if two points belong to the same or different objects, and in fact is the stronger cue for motion boundary perception in human vision. Edge identification is done by comparing the motion vectors at its two vertices.

3.2 Dense Correspondences

The second step performs matching of finer level features with the help of coarsest level matches already identified through perceptual grouping as described above. The finer level features are more densely distributed and therefore they improve the accuracy of detected moving object shapes relative to those segmented at the coarsest level (Fig. 1). A sequence of coarse-to-fine matching steps described in the following paragraph is iterated to the finest level of detected features, yielding the highest density of feature matches. The motions of denser features are predicted based on the known motions of nearby, coarser level features.

The coarser level features near a detected fine level feature may belong to one or more differently moving objects (Fig. 2). Therefore the detected feature may have any of these motions. Accordingly all candidate motions are considered. For a new fine level feature, each of the available motion estimates predicts a different matching location in the next frame. Only one of these is correct and to be selected. Around each such predicted match, feature points are tested to identify those whose neighbors gray levels are well correlated with those of the fine feature point being examined. Since the estimates derived from coarse level are more approximate due to lower feature density, the newly identified candidate matches serve as more accurate alternative motion estimates. Selection among these candidates is now performed by enforcing the spatial continuity of motion.

The unique matched pairs selected for fine level feature points comprise denser correspondences than the coarse level correspondences inherited from the coarser level. These finer level correspondences can again be segmented into distinctly moving objects in the same way as done at the coarser level. The resulting segmentation follows the object boundaries more accurately. These fine level features along with the segmentation are the final result of the iteration. The coarse to fine motion estimation and segmentation is continued at increasingly fine spatial scales until the feature detector no longer gives useful new features.

4 Pixel Correspondences

The method for computation of pixel matches is similar to the method for dense point-features correspondences. Once the finest-level features are found and matched, the remaining pixels are matched using raw intensity information. This is done using intensity correlation, which results in the pixel correspondence for the whole image, as discussed in Sections 4.1 and 4.2, or pairwise image flow, as also computed by the algorithms in [8, 9, 10, 11, 12]. After this step, the motion field is segmented to obtain the boundaries of the moving objects, as discussed in Section 4.3. An attempt is made to integrate the information present in the motion edges and intensity edges to obtain better estimates of the scene structure.

4.1 Finding Candidate Matches for Pixels

Consider a pair of frames (frame 1 and frame 2) in which the finest-level features have been matched. The Delaunay triangulation is computed for the matched points in frame 1 as shown in Fig. 1. The 2D motion of every pixel in this triangle is interpolated from the three vertex motions. The three vertices of a triangle may belong to one, two, or three different moving objects in the scene as shown in Fig. 2. Therefore, all of the pixels in a triangle will have one, two, or three motions for computing the candidates for matching.

As shown in Fig. 3, for each pixel P with coordinates (x, y) and an estimated 2D motion (emx, emy), an $r \times r$ window is selected, centered at the location $(x + emx, y + emy)$ in the next frame. Gray-level correlation is computed at the location $(x + emx, y + emy)$ and at its eight neighboring pixels. The pixels for which this correlation is above some threshold are also considered as candidate matches of P. This is repeated for each possible motion of P to generate all possible candidates for matching. From these candidates, the five with the highest correlations are retained to select the best match in the next step.

4.2 Obtaining the Best Pixel Match

The candidate matches are used by a relaxation algorithm to find the best match for each pixel. The support for a certain candidate match for a pixel is computed

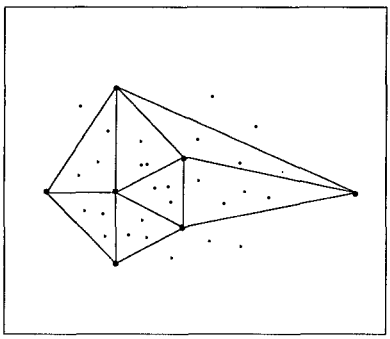


Fig. 1. Vertices of the triangles are the matched, coarse level feature points, and the other dots are new, finer level feature points.

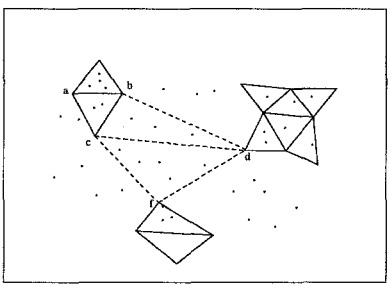


Fig. 2. Triangle *abc* has all vertices from same object, triangle *bcd* has its vertices from two objects and triangle *cdf* has its vertices from three objects.

from the four adjacent pixels, analogous to the Voronoi neighbors in case of feature points. Eventually, two or more pixels may be matched to the same pixel in the next frame. To avoid such situations, the algorithm is used for finding matches from frame 2 to frame 1 also. If a pixel i in frame 1 matches a pixel j in frame 2 while initiating matching from frame 1, then pixel j in frame 2 should match pixel i in frame 1 when matching is initiated in frame 2. Pixel matches not satisfying this two-way constraint are discarded. Near the boundary of moving objects, matches will not be obtained for those pixels corresponding to the scene points that are visible in one frame but not in the other frame. This yields thick bands of unmatched pixels comprising self-occlusion regions of a moving object.

4.3 Segmentation of Moving Objects

The boundaries of the moving objects are obtained based on the similarity of the motion field. However, the detected object boundaries will have errors whenever the motion estimates of pixels are erroneous. This will happen whenever, for example, the number of features in an image part is sparse, leading to rather large triangles. Therefore, the estimates of candidate matches of points within the triangle (for finer-level features or pixels) will contain large errors because

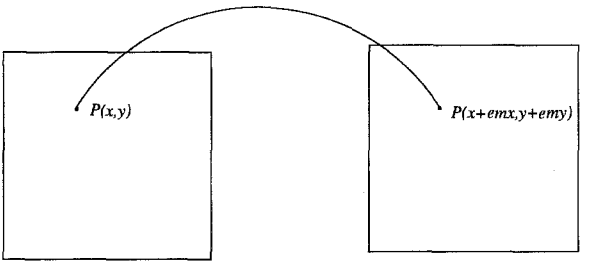


Fig. 3. Pixel P located at (x, y) in frame 1 and its estimated location at $(x + emx, y + emy)$ in frame 2.

the estimates are based on linear interpolation of the vertex motions. This will propagate errors down to both feature matching at the finest scale and pixel matching. It is discussed later as how intensity structure can be used to help overcome some of these shortcomings.

The pixel motion estimates yield a motion field whose discontinuities would ideally correspond to the boundaries of objects moving differently. However, the discontinuities in the estimated motion field are noisy and are jagged versions of object boundaries. The jaggedness is caused by the remaining errors in pixel motion estimates, which are particularly hard to eliminate in the vicinity of the object silhouettes where the object surfaces gradually turn away from the viewer. This is because the image intensity variations of a scene point with motion and the perspective compression of a surface patch are particularly severe under such conditions, resulting in severe correlation-based matching errors.

To address these problems, it is assumed that the object (motion) boundaries coincide with intensity edges. This assumption is valid for a large fraction of real scenes. Consequently, any motion boundary almost coinciding with an intensity edge is interpreted as a noisy version of the intensity edge that is assumed to be the motion and object boundary. Accordingly, the algorithm searches for all intensity edges and tests for similarity of motion estimates on its two sides. If the motion estimates on the two sides are similar, then the intensity edge is merely a marking on the object. If not, the edge is tested as a motion edge.

To this end, intensity edge contours are also obtained which yield regions of similar gray levels. Every pixel is given a label that corresponds to its region. To detect the boundary pixels of the moving objects, a small window centered around an edge pixel is considered. Average motion vectors are computed on the two sides separated by the intensity edge and a measure of similarity between them is computed. If they are not similar then the edge pixel is interpreted as on a motion boundary. By repeating this procedure for all of the intensity edge pixels, the pixels on the motion boundaries of the moving objects are detected.

Another approach to obtain motion boundaries is to merge similar motion regions. For a pair of neighboring regions, average motion vectors are computed for these two regions by considering all the pixels within a small neighborhood of the intensity edge they share.

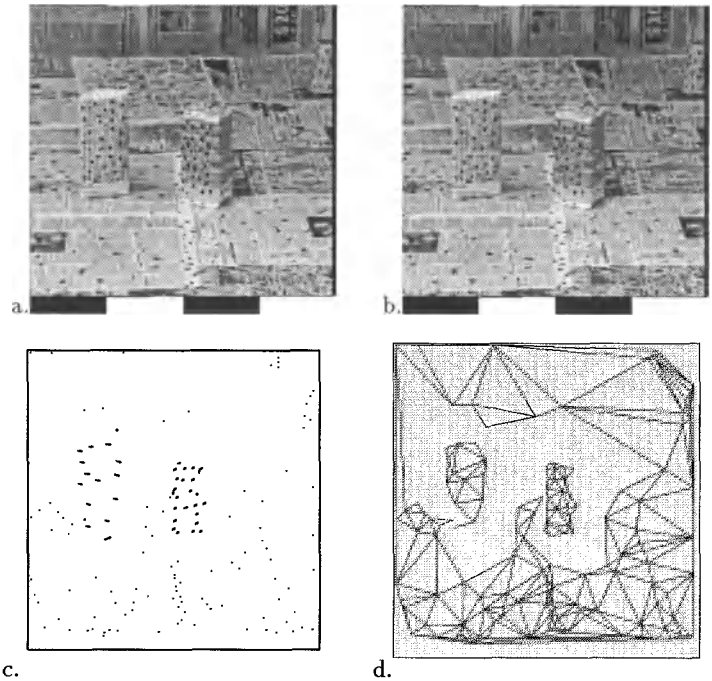


Fig. 4. (a) One frame in an image sequence. (b) Point features detected in the frame. (c) Feature matching at a coarse level. (d) Motion segmentation at the coarse level.

5 Experimental Results

Experimental results obtained by the algorithm presented are shown in this section for two different image sequences. One of these was obtained in our laboratory and the other one shows an outdoor scene.

For the first example a sequence was taken by moving objects with known motion between successive frames as shown in Fig. 4. Two objects are moving. Fig. 4a is the 3rd frame in the sequence and Fig. 4b shows point features detected in this frame. Point correspondences at a coarser level from perceptual grouping between frame-3 and frame-4 are shown in Fig. 4c. The segmentation of moving objects in frame-3 is shown in Fig. 4d. Point correspondences at a denser level are shown in Fig. 5a and segmentation of moving objects is shown in Fig. 5b. Finally, pixel matches are obtained for every pixel in the image and Fig. 5c shows the motion field (for every 5th row and 5th column pixel) between frame 3 and frame 4. Fig. 5d shows the segmentation of motion field.

For the second example the camera is moving while acquiring the image sequence. Fig. 6a is frame 3 in the sequence and Fig. 6b shows the motion field at pixel level.

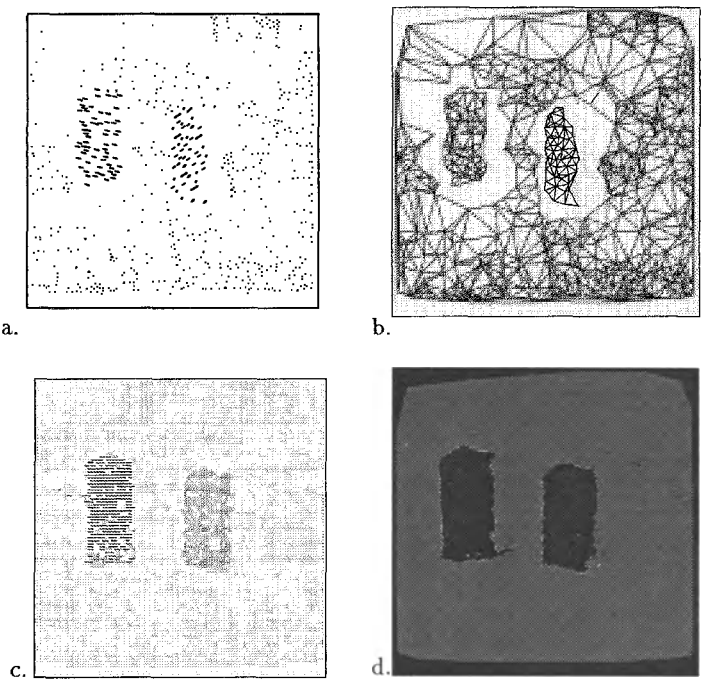


Fig. 5. (a) Feature matching at a dense level. (b) Motion segmentation at the dense level. (c) Motion field (every 5th row and 5th column). (d) segmentation of motion field at pixel level.

6 Conclusions

In this paper, an algorithm is proposed to obtain flow of image points across a sequence. Features as well as pixels are segmented into different moving objects. Experiments were conducted with both laboratory image sequences as well as natural sequences. Intensity maxima and minima were used as features. Other feature point detector may be used to obtain the feature points required for this algorithm.

Perceptual grouping yields reliable correspondences. Although they are sparse, most correspondences found are correct. Inside homogeneous regions, sometimes the pixels are not matched uniquely. When an object also has a motion boundary with little change in texture/gray level across it, the boundary may not be found accurately. We have tried to integrate the intensity edge information with the motion field to more accurately locate the boundaries of the moving objects. Some of the pixels near the boundary of a moving object that are within the stationary background are not matched correctly.

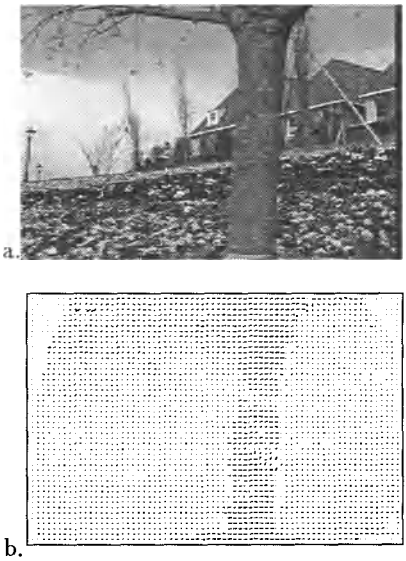


Fig. 6. (a) Frame 3 in the sequence. (b) Motion field (every 5th row and 5th column).

References

1. I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. PAMI-9, January 1987.
2. I. K. Sethi, V. Salari, and S. Vemuri, "Feature point matching using temporal smoothness in velocity," in *Pattern Recognition Theory and Applications* (P. A. Devijver and J. Kittler, eds.), pp. 119–131, New York: Springer-Verlag, June 1986.
3. K. Rangarajan and M. Shah, "Establishing motion correspondences," *CVGIP: Image Understanding*, vol. 54, pp. 56–73, July 1991.
4. C. L. Cheng and J. K. Aggarwal, "A two-stage hybrid approach to the correspondence problem via forward searching and backward correcting," in *Proceedings of the International Conference on Pattern Recognition*, pp. 173–179, 1990.
5. C. Debrunner and N. Ahuja, "Motion and structure factorization and segmentation of long multiple motion image sequences," in *European Conference on Computer Vision*, pp. 217–221, 1992.
6. J. K. Aggarwal and Y. F. Wang, "Analysis of a sequence of images using point and line correspondences," in *Proceedings of the International Conference on Robotics and Automation*, 1987.
7. J. Weng, N. Ahuja, and T. Huang, "Motion and structure from point correspondences: A robust algorithm for planar case with error estimation," in *Proceedings of the International Conference on Pattern Recognition*, 1988.
8. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 236–242, 1992.

9. J. H. Duncan and T. C. Chou, "The detection of motion and computation of optical flow," in *Proceedings of the International Conference on Computer Vision*, pp. 374–382, 1988.
10. D. Heeger, "Model for the extraction of image flow," *Journal of the Optical Society of America*, pp. 1455–1471, 1987.
11. B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.
12. A. Singh, *Optical Flow Computation: A Unified Perspective*. Los Alamitos: IEEE Computer Society Press, 1992.

On the Motion Problem of a Rigid Planar Patch and its Invariance

Chia-Hoang Lee

Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 30050
Republic of China

Abstract. Although theories for recovering the motion and structure of a planar patch are known, existing algorithms are quite sensitive to noise in the image data. This paper presents robust computing procedures for recovering the motion parameters. We introduce two new concepts: (1) a virtual plane that is normal to the translational vector, and (2) the ratio of the depths of a featured point at two different times. These two concepts drastically simplify the analysis of the motion problem and lead to simple and robust procedure for recovering the motion parameters.

1 Introduction

One fundamental task in computer vision is the analysis of a sequence of imaged scenes or objects. A general aim of analyzing such sequences is to recover the structure of the scene/object and the underlying motion it exhibits with respect to the visual sensor.

In this report, our focus is mainly on the feature-based scheme for the case of rigid planar patch. For the general case of rigid object, see [1][2] for further references. In their pioneering work on motion estimation of a rigid planar patch, Tsai and Huang [1] (referred to T-H method below) showed that two views of a patch are sufficient to determine its motion and its structure provided that four features can be identified. However, the computational aspect of the T-H method is quite sensitive to noise in the image data and proves to be satisfactory only if many points are available. Such a least squares approach requires many points to be present and therfore restricts the potential application domains of the technique. In fact, no known computing procedures based on minimum information prove to be robust and the task of searching for a good method is still very illusive.

In this paper, we show that a robust computing procedure based on minimum information can be developed. The major observations of the proposed method consists of two new concepts: (i) The motion of a virtual object that is normal to the translational vector, and (ii) The ratio of the depth of a featured point at one instant to the depth of the same featured point at another instant. The

* This work was supported in part by the National Science Council, R.O.C under grant 81-0408-E-009-565.

ratio is referred to as the quotient depth of the featured point throughout the paper. It turns out that these two concepts drastically simplify the analysis and provide a robust computation for motion recovery.

2 Motion Problem Formulation: Planar Patch

It is well known for a long time in the literature that a 3D rigid motion can be decomposed as a rotation followed by a translation and, in general, be represented as the following equation:

$$y = \mathbf{R}x + \mathbf{T}$$

where x represents the position of a point before 3D motion and y represents the position of the point after exercising the 3D motion. The above equation assumes that the rotation is rotating about an axis with the origin of the coordinate system as the rotational center. It is also known that the rotational center can be chosen arbitrarily in 3D space and the rotational axis can be shifted accordingly. Suppose we choose a point lying on the object, denoted by ref_A, as the rotational center, then the above equation can be rewritten as:

$$\begin{aligned} y &= \mathbf{R}x + \mathbf{T} \\ &= \mathbf{R}(x - ref_A) + \mathbf{R}ref_A + \mathbf{T} \\ &= \mathbf{R}(x - ref_A) + ref_B \\ &= ref_A + \mathbf{R}(x - ref_A) + (ref_B - ref_A) \end{aligned} \tag{1}$$

where ref_B is the corresponding point of ref_A in the second frame. The physical interpretation of the new formula is to view the 3D motion as rotating the object with respect to the reference point and then translate the object so that the new position of ref_A coincides with ref_B. The first two terms of the new formula describe the rotating part while the last term describes the translational vector in the case where ref_A, instead of the origin of the coordinate system, is taken as the rotational center. For the new formulation, the translational vector is equal to $ref_B - ref_A$ instead of the original vector $ref_B - \mathbf{R}ref_A$ and depends on the reference point that is chosen as the rotational center. Although the two formulations (one uses the origin of the coordinate system as the rotational center, while the other one uses a reference point on the object as the rotational center) are, in fact, mathematically equivalent, the computational aspect for obtaining $\mathbf{R}$ and $\mathbf{T}$ is different.

In this paper, we assume that the two perspective views at time t_1 and t_2 , respectively, are taken of a rigid planar patch moving in the 3-D space. The rigid planar patch consists of four featured points, and the task is to derive the motion and structure of the planar patch in the 3D space from the two views. For convenience, the following describes the notations used in this paper. Without loss of generality, the focal length f of the imaging device is taken to be 1. The term of the *ith* point refers to the *ith* point on the rigid object. Let

$z_i A_i = obj_i{}^A$ = Position vector of the ith point at t_1
$z_i' B_i = obj_i{}^B$ = Position vector of the ith point at t_2
$A_i = (x_i, y_i, 1)^t$
$B_i = (x_i', y_i', 1)^t$

where (x_i, y_i) and (x_i', y_i') are, respectively, the image coordinates of the ith point at t_1 and t_2. Then

$$z_i' B_i = \mathbf{R} z_i A_i + \mathbf{T} \quad i = 1..N \tag{2}$$

where

$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{12} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is a rotation matrix, and $\mathbf{T} = [t_x\, t_y\, t_z]^t$ is a translational vector, and z_i, z_i' represents the depths of the ith featured point at t_1 and t_2.

The motion problem is defined as: Given 4 image point correspondences $A_i \leftrightarrow B_i \quad i = 1, 2, 3, 4$, determine rotation $\mathbf{R}$, translational vector T, and the depths z_i, z_i'.

If we choose ref_A as the rotational center, then the formula becomes

$$z_i' B_i = ref_A + \mathbf{R}(z_i A_i - ref_A) + (ref_B - ref_A) \quad i = 1..N \tag{3}$$

where ref_B is the corresponding point of ref_A, and the translational vector $\mathbf{T}$ in the new formulation is $ref_B - ref_A$. Below we will let ref_A be the arithmetic center of the four featured points in 3D space in the section 7 where we shall develop method for deriving $\mathbf{R}$ and $\mathbf{T}$.

3 The Concept of Quotient Depth and Its Computation

In this section, we introduce the concept of ratio, referred to as quotient depth, of the depth of a featured point at the first instant to the depth of the same featured point at the second instant. As the readers shall see, the new concept of the quotient depth has two interesting properties: (1) its computation requires only the cross ratios of simple image measurements, (2) it allows one to compute the depths directly without computing motion parameters in the first place and therefore, to transform "motion problem based on image data" into "motion problem based on 3D data".

Recall that z_i and z_i' denote, respectively, the depths of the ith featured point at two different times. Thus, the quotient depth of ith featured point is z_i/z_i'. Below, the planarity condition of the input data will be used to develop formula for computing quotient depth of each featured point up to a unknown constant k. The constant k, referred to as the base quotient depth, will be defined as the quotient depth of the first featured point, i.e. z_1/z_1'.

Note

$$\mathbf{R}z_1A_1 = z_1'B_1 - \mathbf{T} \tag{4}$$

$$\mathbf{R}z_2A_2 = z_2'B_2 - \mathbf{T} \tag{5}$$

$$\mathbf{R}z_3A_3 = z_3'B_3 - \mathbf{T} \tag{6}$$

$$\mathbf{R}z_4A_4 = z_4'B_4 - \mathbf{T} \tag{7}$$

Since z_1A_1 is coplanar with z_2A_2, z_3A_3, z_4A_4, there exists a_2, a_3, a_4 such that $a_2 + a_3 + a_4 = 1$ and

$$a_2z_2A_2 + a_3z_3A_3 + a_4z_4A_4 = z_1A_1. \tag{8}$$

Applying $\mathbf{R}$ to both sides of (8) and using (4)-(7), it gives

$$a_2z_2'B_2 + a_3z_3'B_3 + a_4z_4'B_4 = z_1'B_1. \tag{9}$$

Dividing both sides of (8) and (9) by z_1 and z_1' respectively, one gets

$$a_2\frac{z_2}{z_1}A_2 + a_3\frac{z_3}{z_1}A_3 + a_4\frac{z_4}{z_1}A_4 = A_1 \tag{10}$$

and

$$a_2\frac{z_2'}{z_1'}B_2 + a_3\frac{z_3'}{z_1'}B_3 + a_4\frac{z_4'}{z_1'}B_4 = B_1. \tag{11}$$

Let $A_{ij} = A_i \times A_j$; $B_{ij} = B_i \times B_j$ where $\times$ denotes vector product, $i, j = 1..4$ and $i < j$.

It is easy to see that

$$a_k\frac{z_k}{z_1}A_k \cdot A_{ij} = A_1 \cdot A_{ij} \tag{12}$$

$$a_k\frac{z_k'}{z_1'}B_k \cdot B_{ij} = B_1 \cdot B_{ij} \tag{13}$$

where $k \neq i, k \neq j$, $i < j$; and $i, j, k = 2, 3, 4$.

Since any three of A_1, A_2, A_3 and A_4 are not collinear, one could divide (11) by (10) and obtain

$$\frac{z_k'}{z_k} = \frac{A_k \cdot A_{ij}}{B_k \cdot B_{ij}}\frac{B_1 \cdot B_{ij}}{A_1 \cdot A_{ij}}\frac{z_1'}{z_1} \tag{14}$$

for $k = 2, 3, 4$, and $k \neq i, j$.

Write out the above formula explictly, one obtains the quotient depth of each featured point as follows:

$$\frac{z_2'}{z_2} = \frac{A_2 \cdot A_{34}}{B_2 \cdot B_{34}} \cdot \frac{B_1 \cdot B_{34}}{A_1 \cdot A_{34}}\frac{z_1'}{z_1} \equiv \delta_2\frac{z_1'}{z_1} \equiv k\delta_2 \tag{15}$$

$$\frac{z_3'}{z_3} = \frac{A_3 \cdot A_{24}}{B_3 \cdot B_{24}} \cdot \frac{B_1 \cdot B_{24}}{A_1 \cdot A_{24}}\frac{z_1'}{z_1} \equiv \delta_3\frac{z_1'}{z_1} \equiv k\delta_3 \tag{16}$$

$$\frac{z_4'}{z_4} = \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \cdot \frac{B_1 \cdot B_{23}}{A_1 \cdot A_{23}} \frac{z_1'}{z_1} \equiv \delta_4 \frac{z_1'}{z_1} \equiv k\delta_4 \tag{17}$$

For notational convience, we will let $z_1'/z_1 = k \equiv k\delta_1$. I.e. $\delta_1 = 1$. Now we will see how the computations of δ_2, δ_3 and δ_4 can be related to the heights of various triangles defined by the image points. From above, we have

$$\delta_2 = \frac{A_2 \cdot A_{34}}{B_2 \cdot B_{34}} \cdot \frac{B_1 \cdot B_{34}}{A_1 \cdot A_{34}} \tag{18}$$

$$\delta_3 = \frac{A_3 \cdot A_{24}}{B_3 \cdot B_{24}} \cdot \frac{B_1 \cdot B_{24}}{A_1 \cdot A_{24}} \tag{19}$$

$$\delta_4 = \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \cdot \frac{B_1 \cdot B_{23}}{A_1 \cdot A_{23}} \tag{20}$$

If we denote the area of the triangle (base) formed by features i, j, k at the first instant as Δ_{ijk} and the area of the triangle at the second instant as Δ'_{ijk}, then we can rewrite the above formula as:

$$\delta_2 = \frac{\Delta'_{134}}{\Delta_{134}} \cdot \frac{\Delta_{234}}{\Delta'_{234}} \quad \delta_3 = \frac{\Delta'_{124}}{\Delta_{124}} \cdot \frac{\Delta_{234}}{\Delta'_{234}} \quad \delta_4 = \frac{\Delta'_{123}}{\Delta_{123}} \cdot \frac{\Delta_{234}}{\Delta'_{234}} \tag{21}$$

Next, consider the triangle defined by features i, j and k and let $h_{i,jk}$ be the height joinning the side defined by j and k and the vertex i. Then it is clear that one can replace the area of each triangle by the height in the above formula since the numerator and denomenator of each quotient share a common side. Therefore, we obtain

$$\delta_2 = \frac{h'_{1,34}}{h_{1,34}} \cdot \frac{h_{2,34}}{h'_{2,34}} \quad \delta_3 = \frac{h'_{1,24}}{h_{1,24}} \cdot \frac{h_{3,24}}{h'_{3,24}} \quad \delta_4 = \frac{h'_{1,23}}{h_{1,23}} \cdot \frac{h_{4,23}}{h'_{4,23}} \tag{22}$$

4 Deriving the Base Quotient Depth k

In the last section, the quotient depths of featured points have been computed up to an unknown constant k, the base quotient depth. The aim now is to describe how the constant k may be related to the second largest eigenvalue of a 3×3 positive definte matrix.

Consider a virtual object Ω, the set of vectors perpendicular to the translational vector $\mathbf{T}$. It is easy to see that the virtual object Ω defines either a plane passing through the origin if $\mathbf{T} \neq 0$, or the whole 3D space if $\mathbf{T} = 0$. From the previous section, $z_2' = k\delta_2 z_2$, $z_3' = k\delta_3 z_3$, $z_4' = k\delta_4 z_4$, and the equations governing the two frames can thus be written as:

$$\mathbf{R} z_1 A_1 = k z_1 B_1 - \mathbf{T} \tag{23}$$

$$\mathbf{R} z_2 A_2 = k\delta_2 z_2 B_2 - \mathbf{T} \tag{24}$$

$$\mathbf{R}z_3A_3 = k\delta_3 z_3 B_3 - \mathbf{T} \tag{25}$$

$$\mathbf{R}z_4A_4 = k\delta_4 z_4 B_4 - \mathbf{T} \tag{26}$$

Taking the scalar products of both sides of equations (23-26) with X in Ω, we obtain

$$X \cdot \mathbf{R}z_iA_i = X \cdot k\delta_i z_i B_i \quad 1 \leq i \leq 4 \tag{27}$$

Factoring z_i out, we get:

$$\mathbf{R^t}X \cdot A_i = kX \cdot \delta_i B_i \tag{28}$$

Let $Y = \mathbf{R^t}X$, $\mathbf{A} = [A_1 \ A_2 \ A_3 \ A_4]$, $\mathbf{B} = [B_1 \ B_2 \ B_3 \ B_4]$ and $\mathbf{M} = (\delta_1 \ \delta_2 \ \delta_3 \ \delta_4)\mathbf{I_{4\times4}}$ where $\mathbf{I_{4\times4}}$ denotes 4×4 identity matrix. Then the above equations can be rewritten in vector form as follows:

$$\mathbf{A}^tY = k\mathbf{M}^t\mathbf{B}^tX \tag{29}$$

Hence

$$(\mathbf{A}\mathbf{A}^t)Y = k\mathbf{A}\mathbf{M}^t\mathbf{B}^tX \tag{30}$$

Since $\mathbf{A}\mathbf{A}^t$ is a nonsingular 3×3 matrix and thus

$$Y = k(\mathbf{A}\mathbf{A}^t)^{-1}\mathbf{A}\mathbf{M}^t\mathbf{B}^tX \tag{31}$$

Let $\mathbf{U_{3\times3}} = (\mathbf{A}\mathbf{A^t})^{-1}\mathbf{A}\mathbf{M^t}\mathbf{B^t}$, then

$$Y = k\mathbf{U}X \tag{32}$$

Since $Y = \mathbf{R^t}X$, thus $Y^tY = X^tX$. From (32), it follows

$$k^2X^t\mathbf{U^t}\mathbf{U}X = X^tX \tag{33}$$

for all X in Ω

Note that $\mathbf{U}$ is a nonsingular matrix as A_1, A_2, A_3 and B_1, B_2, B_3 are non-collinear, respectively. We now show how to compute k in terms of the eigenvalues of the matrix $\mathbf{U^t}\mathbf{U}$. As X is perpendicular to the translational vector $\mathbf{T}$, we may restrict the norm of X to one and rewrite (33) into the following system of equations:

$$k^2X^t\mathbf{U^t}\mathbf{U}X = 1 \tag{34}$$

$$X^tX = 1 \tag{35}$$

$$X \cdot \mathbf{T} = 0 \tag{36}$$

Since $\mathbf{U^tU}$ is positive definite, its eigenvalues, $\lambda_1 \leq \lambda_2 \leq \lambda_3$ are all positive and its corresponding eigenvectors, v_1, v_2 and v_3 can be chosen to be orthonomal. Let $Q = [v_1\ v_2\ v_3]^t$. Then (34) can be rewritten as

$$k^2(QX)^t \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} (QX) = 1 \tag{37}$$

Equation (37) defines an ellipsoid with three axes having $\frac{1}{k\sqrt{\lambda_1}}, \frac{1}{k\sqrt{\lambda_2}}, \frac{1}{k\sqrt{\lambda_3}}$ as their lengths, respectively. Taking equation (35)(36) into account, the solution of the two equations (35-36) is a circle, **C**, on the unit sphere with **T** to be its orientation, and with center at the origin of the coordinate system. Note that **T** is unknown, hence the position of **C** on the unit sphere is unknown. Since points on the unit circle **C** also satisfy eq. (37), **C** lies on the ellipsoid defined by eq. (37). Thus, the problem of computing k is to search for k such that **C** can be cut out of the ellipsoid defined by equation (37).

Without loss of generality, we will transform the ellipsoidal equation into the canonical coordinate system defined by the three axes of the ellipsoid, and call the three axes as x, y, z axis. I.e. we will rewrite the (37) into

$$k^2 Z^t \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} Z = 1 \tag{38}$$

where $Z = (x, y, z)^t$.

Expanding it out, we obtain

$$\lambda_1 x^2 + \lambda_2 y^2 + \lambda_3 z^2 = 1/k^2 \tag{39}$$

Since the circle **C** is a closed curve and symmetric with respect to the origin, there must exist a point $\mathbf{P} = (x_p, y_p, z_p)$ lying on **C** such that $x_p = 0$. Since **P** lies on the unit circle, we have

$$y_p^2 + z_p^2 = 1 \tag{40}$$

Furthermore, the circle **C** also lies on the ellipsoid, we have

$$\lambda_2 y_p^2 + \lambda_3 z_p^2 = 1/k^2 \tag{41}$$

Thus

$$\lambda_2 \leq 1/k^2 \leq \lambda_3 \tag{42}$$

Using symmetric arguments, we can also choose a point $\mathbf{Q}=(x_q, y_q, z_q)$ lying on $\mathbf{C}$ such that $y_q = 0$. And by the same reasoning, we can show

$$\lambda_1 \leq 1/k^2 \leq \lambda_3 \tag{43}$$

Repeating the above argument again, we can choose a point $\mathbf{R}=(x_r, y_r, z_r)$ lying on $\mathbf{C}$ such that $z_r = 0$. And by the same reasoning, we can show

$$\lambda_1 \leq 1/k^2 \leq \lambda_2 \tag{44}$$

Combining (42-44), we obtain $k = \frac{1}{\sqrt{\lambda_2}}$. In other words, k equals to the inverse of the second largest eigenvalue of $\mathbf{U^tU}$, and the following theorem holds.

Theorem: Let $\mathbf{U}$ be defined as above and $\lambda_1 \leq \lambda_2 \leq \lambda_3$ be the eigenvalues of $\mathbf{U^tU}$, then k, the base quotient depth, equals to $1/\sqrt{\lambda_2}$.

5 Computing the depth z_i and z_i'

Let $C_1 = kB_1$, $C_i = k\delta_i B_i, 2 \leq i \leq 4$, then (23-26) can be written as

$$\mathbf{R}z_i A_i = z_i C_i - \mathbf{T}, \quad 1 \leq i \leq 4 \tag{45}$$

It follows that

$$\mathbf{R}(z_i A_i - z_j A_j) = (z_i C_i - z_j C_j), \; 1 \leq i \leq 4 \tag{46}$$

and

$$|z_i A_i - z_j A_j| = |z_i C_i - z_j C_j| \tag{47}$$

Let $p = z_2/z_1, q = z_3/z_1, r = z_4/z_1$, then the following equations hold:

$$\mathbf{R}(A_1 - pA_2) = (C_1 - pC_2), \tag{48}$$

$$\mathbf{R}(A_1 - qA_3) = (C_1 - qC_3), \tag{49}$$

$$\mathbf{R}(A_1 - rA_4) = (C_1 - rC_4), \tag{50}$$

and

$$|A_1 - pA_2| = |C_1 - pC_2| \tag{51}$$

$$|A_1 - qA_3| = |C_1 - qC_3| \tag{52}$$

$$|A_1 - rA_4| = |C_1 - rC_4| \tag{53}$$

$$|pA_2 - qA_3| = |pC_2 - qC_3| \tag{54}$$

$$|pA_2 - rA_4| = |pC_2 - rC_4| \tag{55}$$

$$|qA_3 - rA_4| = |qC_3 - rC_4| \tag{56}$$

Eqs. (51-53) are, respectively, quadratic equations in p, q and r, while eqs. (54-56) are, respectively, homogeneous quadratic equations in (p, q), (p, r), and (q, r). It is easy to compute the solutions for p, q, r and thus z_i.

6 Computing rotation R and translation T: Orthogonal Procrustes Problem

Since the 3D positions of the planar patch at the two instants have been computed and become available in the previous section, the rotation **R** and **T** can then be solved from 3D data. In the section of problem formulation, we have stated that the 3D motion will be taken as a rotation with its rotational axis passing through a reference point on the object followed by a translation. In addition, the reference point, denoted by ref_A, is defined as the arithmetic center of the quadrilateral planar patch. Obviously, the corresponding point of ref_A, denoted by ref_B, can be found by computing the arithmetic center of the quadrilateral planar patch at the second instant. Since the rotation has no effect on the position of the reference point ref_A, the positional difference between ref_B and ref_A accounts for the translational vector. Thus

$$\mathbf{T} = ref_B - ref_A = (\sum z_i' B_i)/4.0 - (\sum z_i A_i)/4$$

Now consider the problem of deriving rotational matrix **R**. It is clear that the rotational matrix **R** maps ${obj_i}^A - ref_A$ to ${obj_i}^B - ref_B$ for $1 \leq i \leq 4$ and interesting to know that the problem of deriving the best estimate for **R** in the current context is in fact known as an orthogonal procrustes problem [9] in numerical mathematics. The followings formulate the problem: Let P and Q be data matrices with ${obj_i}^A - ref_A$ and ${obj_i}^B - ref_B$ as the ith column, respectively, then the optimum estimate for **R** can be obtained by solving:

$$minimize ||\mathbf{R}P - Q||_F \ subject\ to\ \mathbf{R}^{\mathbf{t}}\mathbf{R} = 1$$

where $|| \ ||_F$ denotes the Frobenius norm.

For reader's convenience, we will use the algorithm [9] which solves the above problem as follows:

Algorithm

1. Let $C = PQ^t$
2. Compute the SVD $U^tCV = \Sigma$. Save U and V
3. Then $\mathbf{R} = VU^t$

7 EXPERIMENTAL RESULTS

All experiments assume the field of view of the camera is 60° and the object size subtends about 20° to 30°. The image is 512×512 pixels. Although extensive simulations were performed, we only list one set of the results in the Table 1. The tilt, slant and the rotational angle of the motion parameters used in the Table 1 are all ten degrees. The first column of the Table lists the range of noise. For instance, three pixels means that noise ranging from 0 to 3 pixels is uniformly added to the horizontal and vertical positions of the data independently. Thus the maximum error between the accurate data and the noisy data is $3 \times \sqrt{2} \simeq 4$

pixels. The largest noise in the last row is about 4 pixels. In the second column, we list the mean relative error in terms of translation vector. In the third column, we list the mean error over 400 experiments. For instance, the result of the first row of Table 1 says: If the true $\mathbf{R}$ and the derived $\hat{\mathbf{R}}$ are applied to any vector x, then the angle between the resulting two vectors will not exceed 3.1 degrees on average.

8 Concluding Remarks

Two new concepts have been introduced to facilitate the image sequence analysis: (1) the ratio of depths over time for every featured point in two scenes and (2) the motion of a virtual plane normal to the translation vector. Interestingly, these two concepts not only simplify the analysis drastically but also provide a robust computing procedure for motion recovery.

References

1. Tsai, R. Y. and T.S. Huang (1982), "Estimating Three-Dimensional Motion parameters of a Rigid Planar Patch II: Singular Value Decomposition", IEEE Trans. Acoustic, Speech and Signal Processing, Vol. ASSP-30.
2. H.H. Nagel and B. Neumann, "ON 3-D Reconstruction from Two Perspective Views", *Proc. IJCAI* 81, Vol. II, Aug. 1981.
3. J.K. Aggarwal, "Motion and Time-Varying Imagery- An Overview", Proc. of IEEE Workshop on Motion: Representation and Analysis, 1986.
4. Bellman R. (1960), *"Introduction to Matrix Analysis"*, McGraw-Hill, New York. p. 25.
5. Strang, G. (1980), *"Linear Algebra and Its Applications"*, Academic Press, New York.
6. Parlett, B (1980), *"The Symmetric Eigenvalue Problem"*, Prentice-Hall, New Jersey.
7. J. Fang and T.S. Huang, "Some Experiments on Estimating the 3D Motion Parameters of a Rigid Body from two Consecutive Image Frames" *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol PAMI-6, NO. 5. 1984.

Noise level	Translation	Angle
one pixel	0.08%	0.9
two pixels	0.60%	2.1
three pixels	1.0%	3.1

Virtual Space Teleconferencing: Real-Time Detection and Reproduction of 3D Face and Body Images

Fumio Kishino, Kazuyuki Ebihara and Jun Ohya

ATR Communication Systems Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
kishino{ebihara, ohya}@atr-sw.atr.co.jp

Abstract

This paper describes a virtual space teleconferencing that proposed as an human-oriented telecommunication system. By using this system, participants at different sites can engage in a conference with the sensation of sharing the same space, and cooperative work can be performed among remotely located participants. More specifically, our system connects three different sites via a 1.5 Mbps ISDN now in commercial use. The system has two large screens that can provide real-time reproduction of 3-D whole body human images. Participants at the three different sites are able to feel as if they are all at the same site.

In this experimental system, to achieve real-time reproduction, tape marks, a data glove and magnetic sensors are used. However, these tools are not appropriate for natural human communications. It is a final goal to develop a system that does not require such tools. We also describe a new facial expression detection method that does not need the tape marks, and real-time detection is realized by exploiting frequency domain data from the face image acquired by the CCD camera. Experiments show very promising results for accurate expression detection and for implementation of real-time hardware of the using propose method.

1. Introduction

The coming visual communications technologies are expected to provide real-world sensations. To achieve real-world sensations, it is essential to reproduce the 3-dimensional surroundings of remote sites around a receiver. Recently, virtual reality (VR) or artificial reality (AR) has been progressing worldwide. This technology gives humans the sensation of being involved in a virtual environment. For natural human-human communications, it is essential not only to assess the situation at each remote site but also to transmit the receiver's status to the remote sites. If it were possible to truly create a realistic virtual world precisely modeled on the real world by using computer vision (CV) and computer graphics (CG) technologies and if it were also possible to create virtual human beings as receivers, such a virtual world could be shared and used for two-way or multi-point communications, making users feel as if they were truly coexisting in the same physical space.

This paper describes a virtual space teleconferencing system that achieves communications with realistic sensations. First, we introduce, three-point virtual space teleconferencing system built to show our concept and to evaluate our proposed system. Next, we describe real-time detection and reproduction of 3D human images that requires participants to wear much equipment. Then we propose a new real-time facial expression detection method that does not require the

tape marks. The proposed method exploits frequency domain data from the face image acquired by the CCD camera for the realization of real-time expression detection.

2. Virtual Space Teleconferencing and Its Experimental System

In order to achieve "Teleconferencing with realistic sensations"[1], the concept of virtual space teleconferencing was examined. As shown in Figure 1, a "virtual space teleconferencing" system creates an image of a conference room (virtual conference room) using computer graphics in real time. It also constructs images of the remotely located conference participants. Users of the system can talk to each other or participate in a conference with the sensation of "sharing the same space". It has the following features.

-The system enables participants to see what they want to see from the angle of their choice, just as in a real space, on a screen.

-All remotely located participants can freely manipulate any object shown on the screen using hand gestures and speech recognition. Still, there are many problems to be solved before this system can be made practical.

This paper focuses on real-time detection and reproduction of 3D face and body images, which is a key requirement for achieving the proposed system.

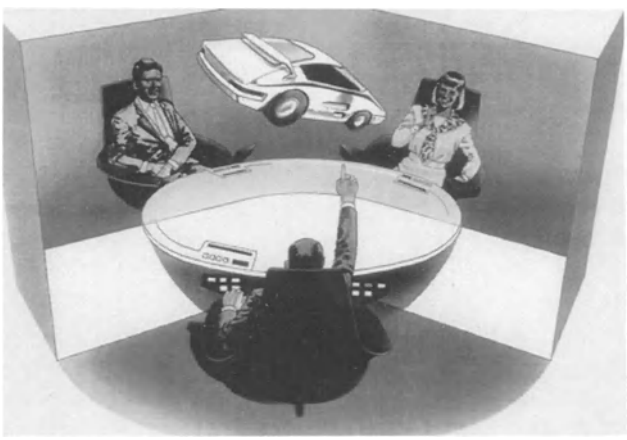


Fig. 1 Virtual Space Teleconferencing Concept

The experimental system was established by connecting three different sites. This system was demonstrated at ITU-PP '94 in Kyoto from September. 19-22, 1994. Figure.2 shows the system configuration of ATR's virtual space teleconferencing experimental system. The main terminal is located at the Kyoto International Conference Hall (KICH), and the other terminals are at ATR. KICH and ATR are connected through a 1.5 Mbps ISDN line. The system has the following features.

(1) Making CG images of people highly qualitative

We applied a model-based coding approach using recognition and image reproduction of a three-dimensional human body model. The technologies will be shown in detail in the next section. This approach it quite possible for one to stand up and look at something from a higher angle focused just right for the viewer. How accurate this CG could be depends upon how accurate the camera could be as a sensor in measuring the facial expressions of the participants and changing the shape of them according to the measured data. More realistic movements and facial expressions could be seen on the screen as a result of our efforts to be more accurate, for example, measuring the whole body including the lower part of the body and putting more marks on the face.

(2) Making the manipulation of objects easier by detecting collisions and using speech recognition results

When three remotely located people work cooperatively, discussing how to built a complicated object like a portable shrine, they not only use hand gestures but also use collision detection and speech recognition. They are able to detect the collisions between the triangular polygons of virtual objects in real time. The system gives the users two types of feedback, one is the sound of the collision(s) and the other is the color change of the collided surfaces. With this aid it is possible, to a certain degree, to infer the spatial relations of objects, even if they are occluded.

As a result we found that cooperative work becomes easier when the sounds of collisions can be heard and the origin of the sounds can be determined.

(3) Display of complicated pictures in a virtual space

Eye and hand motion as well as body gestures are important factors for human communications. Therefore, it is important to display participants as close as possible to their real size, so that a higher degree of reality can be achieved. To accomplish this, we have developed a large wide-field-of-view display. The display combines two 70-inch back-projected screens connected at 120 degrees in a seamless manner. Objects and participants in the display can be looked at from different directions and distances by a participant who wears a pair of LCD glasses fitted with a magnetic position sensor. The ability to look in 3-D at objects and other participants from different perspectives in this wide-field -of-view display gives the participants the impression of sharing a much larger environment, thus increasing the feeling of presence.

(4) Low bit rate transmission

The proposed system displays 500x2500 pixels on a screen at 120 frames per second, and the transmission data is greater than that of HDTV. We confirmed that all data can be transmitted using 1.5 Mbps ISDN lines between our experiment. The reason transmission of such low bit rates is possible is that this system does not transmit the pictures themselves but the data measured with the sensors, and workstations on the display can make CG images on a real-time basis. In this age of B-ISDN, a low bit rate is still considered very important.

Participants not only manipulate interactively, but also generate and modify 3D shapes with verbal descriptions, aided by simple hand gestures. Figure.3 shows a scene of three-point virtual space teleconferencing.

Most of the participants agreed that the 3D human image reproduction was quite natural and smooth, but further study is necessary for a more accurate image quality evaluation.

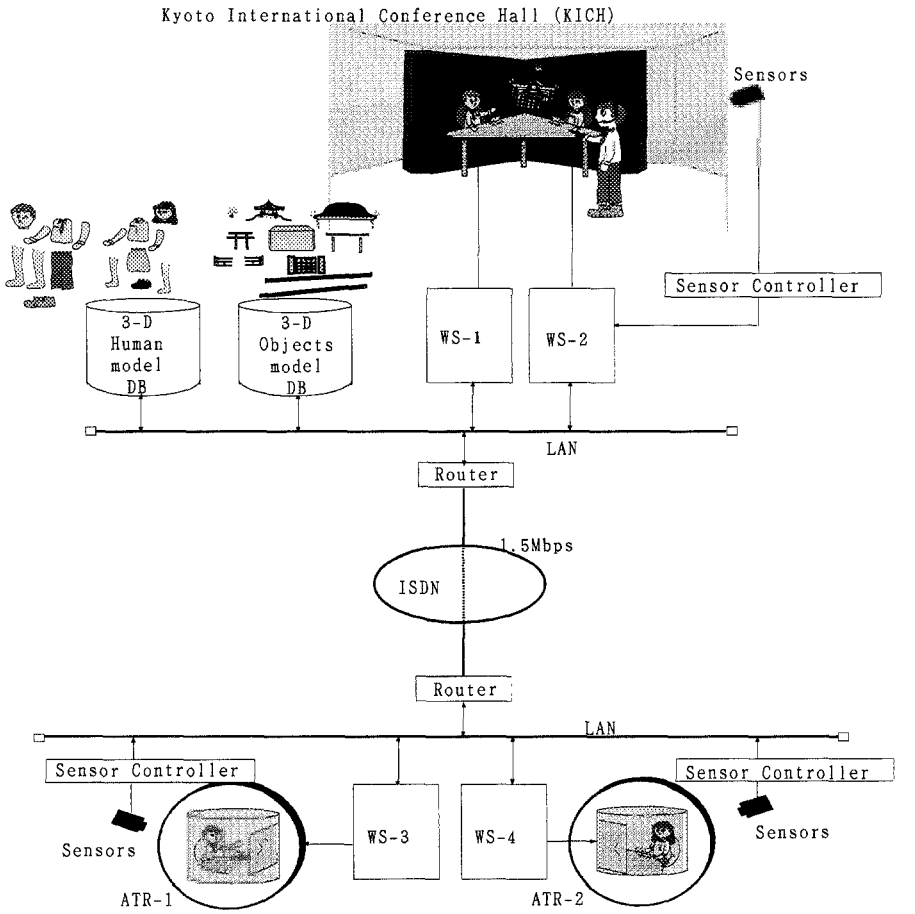


Fig. 2 System Configuration of Three-point Virtual Space Teleconferencing

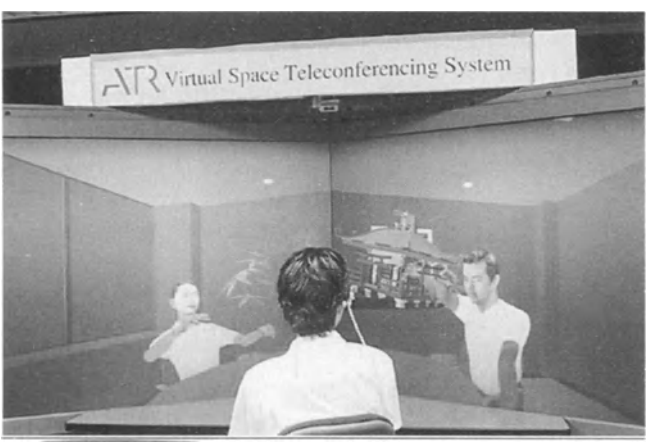


Fig. 3 Three-point Virtual Space Teleconferencing

3. Reproduction of 3D Human Images

A method for human image recognition and reproduction is one of the most important research problems to be solved. In this system, perspectives of human images from the viewpoints of participants must be synthesized, and should include correct shades and shadow to suit the lighting conditions of the virtual room. To solve these problems we applied a model-based coding approach using image recognition and image reproduction of a three-dimensional human body model.

An experimental system enabling real-time reproduction of 3D human images was developed to evaluate the effectiveness of virtual space teleconferencing[3]. An experimental system of real-time detection and reproduction of 3D human images consists of three modules: 3D modeling of the human body, real-time detection of body motions and real-time synthesis of human images. 3D modeling of the human body is done using a Cyberware color 3D digitizer, which rotates around an object, projects laser stripes, and acquires color information and 3D coordinates of each point on the surfaces of body parts. The acquired 3D coordinate data is transformed into a 3D wire frame model consisting of triangular patches. More specifically, if the difference in the directions of surface normals to neighboring points is smaller than the threshold, the points are merged to form an area with a uniform direction. After this process is applied to all points, each area is divided into triangular patches. Then, the color information of each point is mapped to its corresponding triangular patch. 3D information is acquired by dividing the head, body, arms and legs. The 3D model of each part is articulated with another.

It is very difficult to achieve real-time detection of body motion with most image processing technologies. Therefore, to detect the positions of the head, hands and body in real-time, magnetic sensors, which can detect the three translation and rotation parameters, are used. Finger movements are detected by using a data glove. For real-time detection of facial expressions, thirteen blue tape marks are attached to the facial muscles that strongly affect changes in facial expressions(Fig.4).

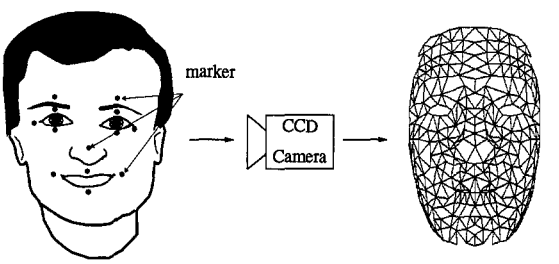


Fig 4 Expression Reproduction Method of the Current System

The 3D coordinates of the nodes in the head, hand, finger and body areas are obtained from the information detected by the magnetic sensors and data glove. The results from tracking the marks on the face are used to move the nodes of the wire frame model for the face. In this experimental system, to achieve real-time reproduction, tape marks, a data glove and magnetic sensors are used. However, these tools are not appropriate for natural human communications. Arideal future goal is develop a system that does not require such tools.

4. Real-time Facial Expression Detection without Markers

Tape marks are not appropriate for natural human communication. Image processing based methods are good due to their passive nature, and there have been some works on facial expression detection(e.g.[4]). However, real-time expression detection from face images has not yet been achieved.

In this section, we propose a new real-time facial expression detection method that does not need tape marks. The proposed method exploits frequency domain data from the face image acquired by a CCD camera for the realization of real-time expression detection. In the following subsections, more details of the method and experimental results are described.

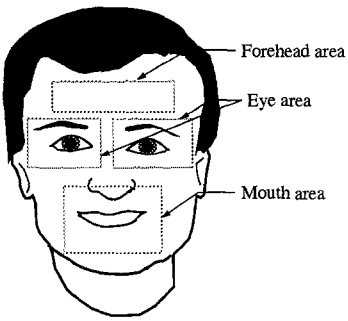


Fig.5 Windows for Detecting Expressions

4.1. Outline

Face images of a person are acquired by a color CCD camera fixed to a helmet worn by the person. As shown in Figure.5, four windows are applied to four areas of the face: the left and right eyes, the mouth and the forehead. Information to be detected includes: (1) in the eye windows, how the eyes are opened, (2) in the forehead window, wrinkles, and (3) in the mouth window, how the mouth is opened.

The distances between the four areas depend on the person. In our method, since frequency domain data from each window are used, frequency components are not affected by positional variations of the face components as long as the components remain in their corresponding windows. In addition, rotations of the helmet usually do not result in fatal errors.

4.2 Algorithm

Figure 6 shows a block diagram of the proposed method.

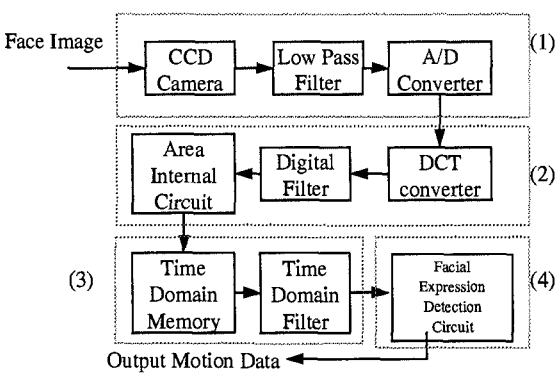


Fig. 6 Block Diagram

(1) Preprocesses

The images acquired by the camera consist of the NTSC signal with an approximately 4.2 MHz bandwidth. Usually, high frequency components in face images correspond to noise caused by slight unevenness in facial skin. To eliminate the noise components, high frequency components of over 2.8 MHz are suppressed by a low-pass filter (LPF). In the subsequent processes, low pass filtered intensity information (Y signal) is used.

(2) Frequency domain transform

Many techniques have been proposed for frequency domain conversions[5]. In this paper, we use DCT (Discrete Cosine Transform) for the conversion because IC chips for DCT have already been developed for real-time implementations on coding methods such as JPEG and MPEG .

The window of each facial area in Figure. 5 is divided into blocks, and each block is transformed by DCT into the domain of spatial frequency as indicated in the following equation.

As shown in Figure.7 the block size M=N=8(pixels) is used in this paper. The domain of spatial frequency consists of the DC component and AC components as illustrated in Figure.7.

In this paper, since actions of the facial components can basically be decomposed into the horizontal, vertical and diagonal directions, we use DCT features for the three directions as described below. Unnecessary components are suppressed by applying the filter to the frequency domain. The features from the horizontal, vertical and diagonal directions are obtained from adding the components in the areas that are defined based on the Q-table[5] .

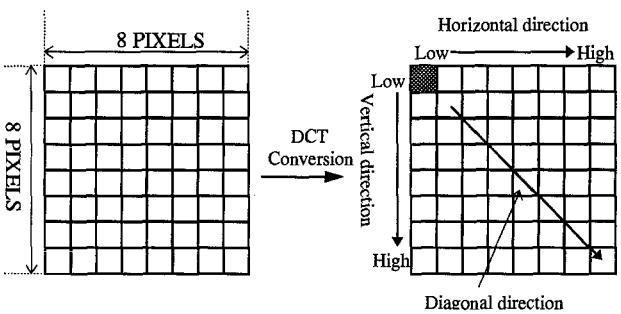


Fig. 7 DCT Transform for Each Block

(3) Facial expression detection

After suppressing unnecessary components by applying the filter to the frequency domain, a three dimensional vector is obtained in each block, where the elements of the vector are the summations of the energies over the components in the three directions. In each window, the three dimensional feature vector whose elements are the summations of the three elements of the vectors from the blocks in the window is obtained.

Examples of the relationship between the feature vectors and facial expressions are as follows. In the eye window, closing the eye decreases the value of the horizontal element and increases the vertical element, while opening the eye widely increases the horizontal element and decreases the vertical element (Fig.8). Similarly, in the forehead window, wrinkles increase the vertical element. The three element values of the feature vector in each window are related to by a conversion table real 3D movements of each facial component. by a conversion table.

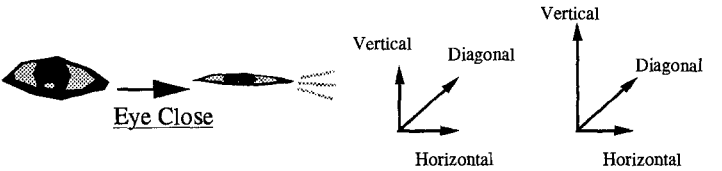


Fig. 8 Eye Expression Detection

4.3 Experimental results

The process speed of the proposed method was studied with a simulated circuit, and it was found out that all processes except the time domain processes finish within the interval between two successive frames.

Figure 9 shows the feature values of the four windows from the time-sequential images, where the subject's expressions change from neutral to "upraise" via brief blinking. The time lag due to the time domain processes as well as the other processes is only 90 ms by the simulated circuit. This is a very promising result for real-time performance of the proposed method. As shown in Figure.9, the difference in the openness of the left and right eyes at 'b' are correctly detected.

Effectiveness of wrinkle detection in the forehead and eyes(brow) is demonstrated in Figure.10. The horizontal wrinkle in the forehead, disappearance of wrinkles, and the vertical wrinkles in the brows can be easily identified at 'a', 'b' and 'c', respectively.

To realize an IC for the proposed method, the circuit scale other than DCT will be only approximately 50,000 gates. It is quite easy to manufacture the IC by current technologies.

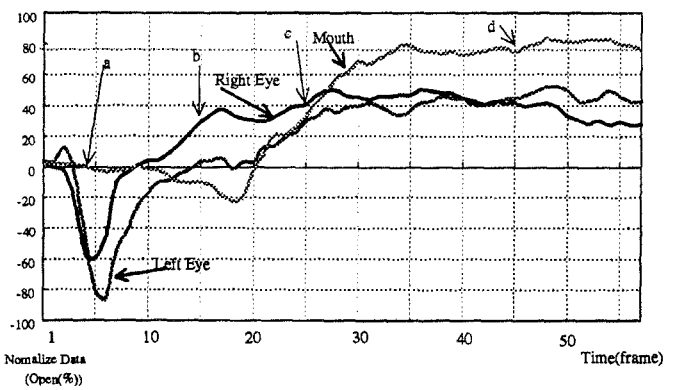
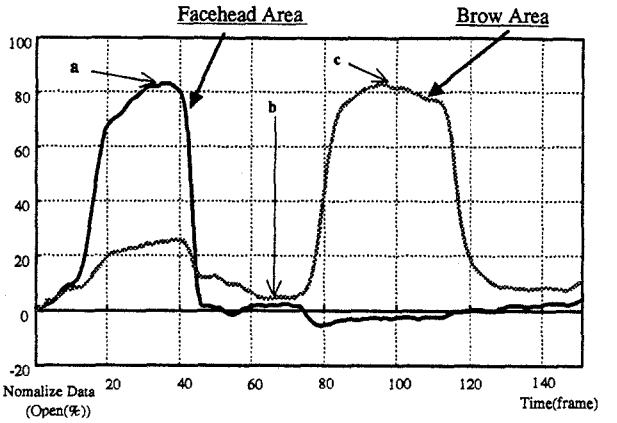


Fig. 9 Facial Expression for "Surprise" via Brief Blink



Expression at 'a' Expression at 'b' Expression at 'c'

Fig. 10 Wrinkles

5. Conclusions

Technologies described here to achieve communications with realistic sensations, or in other words, virtual space teleconferencing, have focused on real-time detection and reproduction of 3D face and body images. We have proposed a new facial expression detection method that does not require tape marks. Our system still forces participants to wear special equipment, namely LCD shutter glasses and sensor-fitted gloves, that decreases the feeling of realism we want to achieve. At ATR we are investigating techniques that solve these issues. A stereoscopic viewing technique using lenticular screens that does not require the use of special glasses is an example. Real-time hand gesture recognition using a 3D prediction model, which does not need any special gloves, is also a subject of on going research at ATR.

Virtual space teleconferencing incorporates a high degree of reality with an interface to support a cooperative work space. Using our system, the interacting participants can fully communicate on verbal information among themselves (i.e., gaze, facial expressions, body postures, hand gestures and so on). At the same time, they can combine hand gestures with verbal descriptions to perform operations on virtual objects.

The technical advances being made will not only provide applications for virtual teleconferencing but also for human-computer interaction, tele-existence systems and other systems that require a natural, realistic visual communication environment.

References

[1] Harashima H. and Kishino F.: "Intelligent Image Coding and Communications with Realistic Sensations -Recent Trends", Trans. IEICE, E74,6 (1991)

[2]Fumio Kishino, Tsutomu Miyasato and Nobuyoshi Terashima,Virtual Space Teleconferencing "Communication with Realistic Sensations",IEEE'95,Robot and Human Communication Processing'95

[3] J. Ohya, Y. Kitamura, H.Takemura,H.Ishii, F. Kishino and N. Terashima, "Virtual Space Teleconferencing: Real-Time Reproduction of 3D Human Images", JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION Vol.6,No.1,March,pp.1-25,1995

[4] Y.Kitamura, Y.Nagashima, J.Ohya and F.Kishino,"Facial image synthesis by hierarchical wire frame model", SPIE'92,Visual Communication Image Processing '92.

[5] " Digital compression and coding of continuous-tone still images: requirements and guidelines", ISO/IEC 10918-1.

Active Vision

High-Performance Tracking with TRICLOPS

Albert J. Wavering[1], Henry Schneiderman[1,2], John C. Fiala[3]

[1] National Institute of Standards and Technology, Intelligent Systems Division
Gaithersburg, MD 20899
wavering@cme.nist.gov
[2] Carnegie Mellon University, Robotics Institute
Pittsburgh, PA 15213
[3] Boston University, Department of Cognitive and Neural Systems
Boston, MA 02215

Abstract. This article reviews the development of experimental visual tracking algorithms which have been implemented using TRICLOPS (The Real-time, Intelligently-ControLled Optical Positioning System), a high-performance active vision system designed and built at the National Institute of Standards and Technology (NIST). These algorithms range from triangulation using simple targets for very high-speed tracking, to model-based techniques which allow robust tracking of polyhedral targets amidst cluttered backgrounds. In all cases, effective prediction of the target motion to compensate for image processing delays is critical to achieving responsive tracking.

1 Introduction

The goal of visual tracking is to keep a moving target centered in the field of view of one or more cameras. This capability is useful for such applications as robotic assembly, surveillance, and autonomous driving/navigation. In order to support visual tracking and other active vision research at NIST, a trinocular robot head, called TRICLOPS, was designed and built [1]. This head has been used for extensive experimentation with a variety of tracking algorithms.

In this paper, we will first summarize the design and performance characteristics of the TRICLOPS active vision system. This is followed by a discussion of tracking algorithms which have been developed to perform high-speed tracking of simple visual targets. Finally, a brief synopsis of work performed using TRICLOPS for model-based target tracking in a complex visual environment is presented.

2 TRICLOPS Active Vision System

One of the primary design goals for TRICLOPS was to build a system with very high dynamic performance—on a par with the human oculomotor system. This emphasis on dynamic performance, along with other requirements such as high accuracy and durability, lead to the development of TRICLOPS as the first direct-drive robot head (Figure 1). The frameless DC motors and resolvers mounted directly to each of the motion

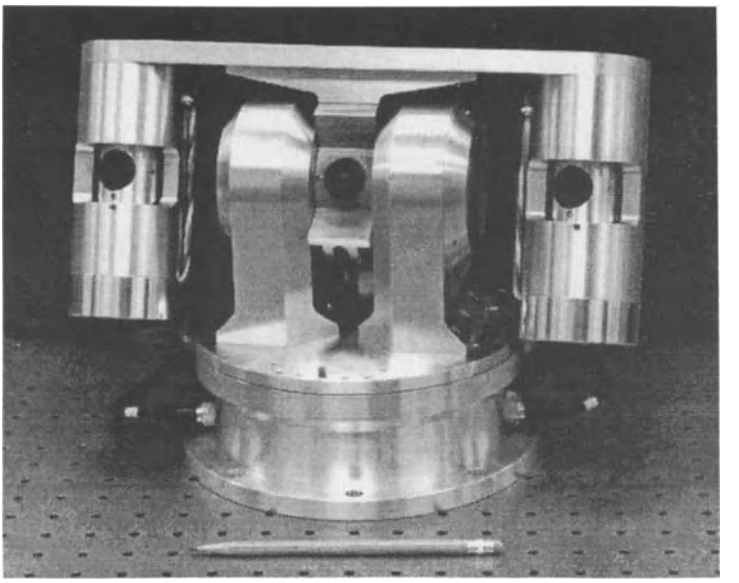


Fig. 1. TRICLOPS active vision system.

axes deliver large accelerations with low friction. Transmission backlash and compliance, which cause errors and oscillations, are effectively eliminated.

TRICLOPS has four mechanical degrees of freedom, as shown in Figure 2. The kinematic arrangement of the four axes is as follows: 1) pan (or base rotation) about a vertical axis through the center of the base, 2) tilt about a horizontal line that intersects the base rotation axis, and 3) left and right vergence axes which intersect and are perpendicular to the tilt axis.

Visual sensing for TRICLOPS is provided by three "micro-miniature" video cameras; one in each of two independently-controlled vergence units, and a third center camera which shares the base pan and tilt motions of the vergence cameras. Equipped with a wide angle (3-4 mm focal length) lens, this color camera may be used to identify general areas of interest in the wide field of view. The longer focal length lenses (15-24 mm) of the vergence cameras allow a more detailed look at specific objects or features. The combination of the center and vergence cameras thus provides the capability to view a scene at multiple resolutions simultaneously, providing a (much simplified) version of the multiresolution foveal-peripheral vision found in many biological visual systems. TRICLOPS is currently configured with a 4 mm center lens and 15 mm vergence lenses.

The performance specifications of TRICLOPS are summarized in Table 1. Further details regarding the features, design, and construction of TRICLOPS may be found in [1].

A multiprocessing controller based on the NASREM Architecture [2] has been constructed to control TRICLOPS and to perform sensory processing and world modeling operations on camera image data. The NASREM Architecture is a hierarchical

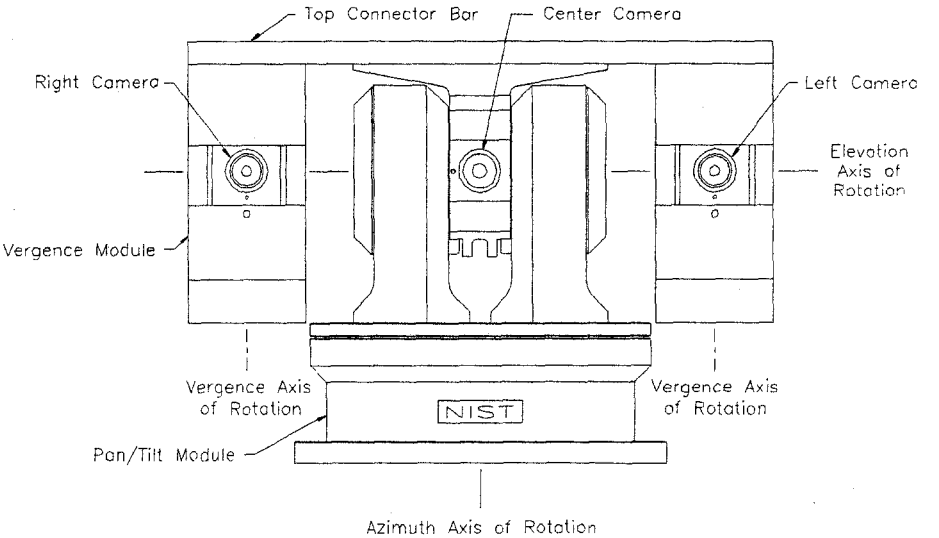


Fig. 2. TRICLOPS degrees of freedom.

Table 1: TRICLOPS Specifications.

Range of motion:	
Pan	+/-1.68 rad (+/-96.3 deg)
Tilt	0.48 rad down, 1.14 rad up (+27.5 deg, -65.3 deg)
Vergence	+/-0.77 rad (+/-44 deg)
Peak acceleration:	
Pan	70 rad/s^2 (4010 deg/s^2)
Tilt	320 rad/s^2 (18,300 deg/s^2)
Vergence	1100 rad/s^2 (63,000 deg/s^2)
Peak velocity:	
Pan	11.5 rad/s (660 deg/s)
Tilt	17.5 rad/s (1000 deg/s)
Vergence	32 rad/s (1830 deg/s)
Slew time:	
Pan (3.14 rad move)	0.523 s
Tilt (1.55 rad move)	0.167 s
Vergence (1.5 rad move)	0.091 s

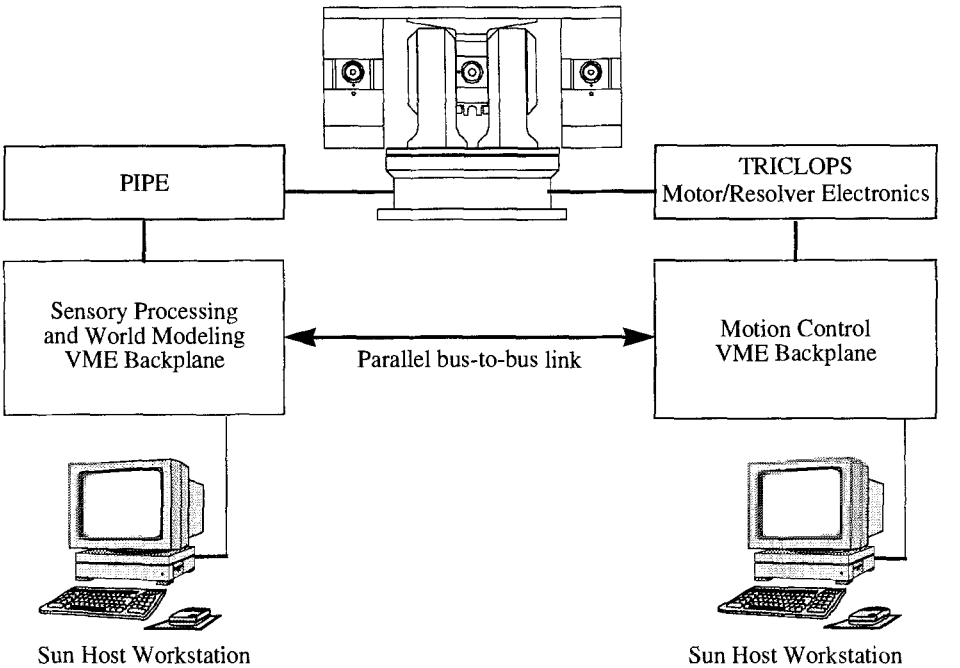


Fig. 3. TRICLOPS control and image processing hardware.

control system architecture which divides the sensing and control problem into discrete levels. At the lowest level are the servo control loops for the device actuators; at higher levels, groups of sensors and actuators are coordinated as equipment subsystems. The NASREM hierarchical control system provides a modular, easily-modifiable infrastructure for implementing and testing active vision algorithms.

The processes and operation of the TRICLOPS motion control and image processing systems are described in [1]; a brief summary is presented here. The full potential of an active vision system cannot be realized without real-time image processing capability. In our lab, this capability is provided by a Pipelined Image Processing Engine (PIPE)[1] for low-level image processing, along with a multiple-cpu VME system for higher-level sensory processing and world modeling (Figure 3). The motion control portion of the TRICLOPS control system architecture consists of a Primitive Level for producing coordinated motions of all axes, and a Servo Level for servoing the axes to commanded joint positions. The Primitive and Servo processes are implemented on five processor boards in a separate motion control VME system, also shown in Figure 3. Taking advantage of the most recent advances in high-speed microprocessors

1. Commercial equipment and materials are identified in this paper in order to adequately specify the experimental procedure. Such Identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily the best for the purpose.

and digital signal processing (DSP) hardware would result in a substantial reduction in the number of processor boards required for the system.

3 Fast Tracking of Simple Targets

One of the most basic behaviors desired of an active vision system is to track an object, using visual information to direct the gaze and keep the object in the field of view of the cameras. In this section, we discuss tracking algorithms which emphasize maximizing the tracking performance in terms of speed (bandwidth). Further details regarding the algorithms and experimental results presented in this section may be found in [1], [3], and [4].

To perform high-speed tracking, image processing delays must be minimized as much as possible. Consequently, the image processing used for these tracking experiments consists of computing the centroid of a thresholded intensity image of a single object which presents a high-contrast image to both the left and right vergence cameras. For these experiments, only the two vergence cameras are used. To create a simple visual scene, cellophane film is used to block out visible light, along with an incandescently-illuminated spherical target. The infrared emission of the incandescent source passes through the filter, while the ambient fluorescent lighting is filtered out, resulting in a nearly ideal image.

For this high-speed tracking, 3-D world position-based modeling and prediction of the target motion is used to generate goal fixation points. A block diagram of the tracking control system is presented in Figure 4. New centroid data are obtained from the world model every 1/30 s. Delayed joint positions which correspond in time to the centroid data are also read in at this time. The amount of delay is equivalent to the image acquisition and processing latency (about 0.084 s in this case), plus the effective delay resulting from interpolation between target position updates (0.033 s). The centroid information is corrected for lens distortion, and the corrected centroids are used along with the delayed joint positions to compute the position of the object with respect to the TRICLOPS world reference frame using a triangulation algorithm.

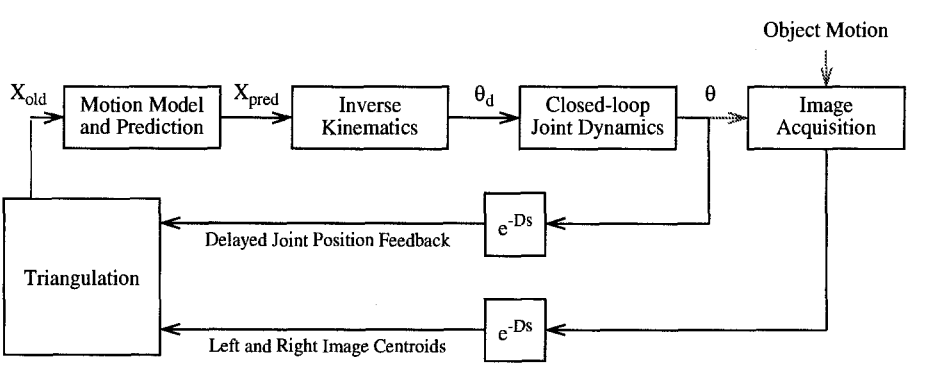


Fig. 4. Block diagram of tracking algorithm.

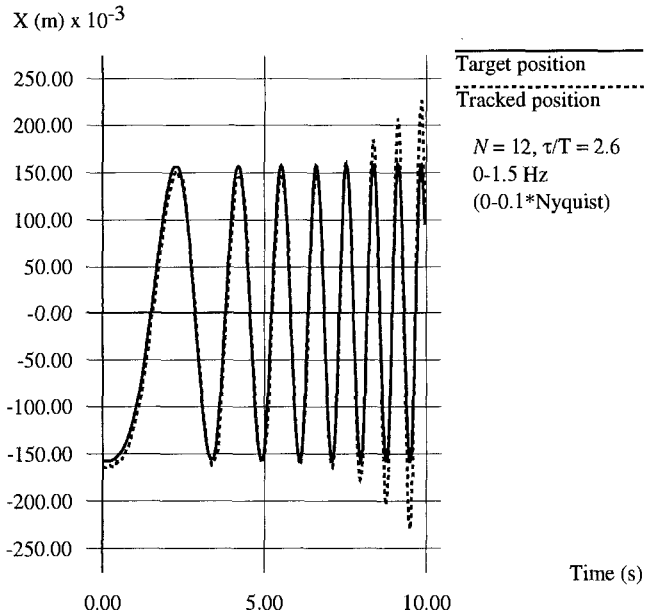


Fig. 5. Tracking performance with cubic LMSF filter.

A number of different types of predictive filters have been used for motion modeling and prediction, including constant coefficient Kalman (α–β–γ) filters and polynomial least-mean-square-fit (LMSF) filters. The equivalent transfer function forms of these filters are derived, and their frequency response characteristics are discussed, in [3].

The tracking performance of TRICLOPS using the system of Figure 4 with a 12-sample cubic LMSF predictive filter is shown in Figure 5. To obtain these tracking results, the redundancy of TRICLOPS (4 degrees of freedom for a 3 degree-of-freedom pointing task) was used to create apparent motion from a stationary target. A fixed target was visually tracked using the tilt and vergence axes while a sinusoidal motion of the TRICLOPS base rotation axis was performed. The frequency of the base motion was increased linearly with time, and the magnitude of the commanded base motion was +/-0.175 rad (+/-10 deg). Figure 5 shows the actual horizontal position of the target and the corresponding tracked position (with respect to the moving base coordinate system).

The tracking performance exhibited in Figure 5 is determined almost entirely by the frequency response of the predictive filter. This filter allows tracking without overshoot up to about 1 Hz. Beyond this frequency, the rise in the magnitude response of the predictive filter results in tracking errors. The 40% overshoot in the tracking response at 1.5 Hz is attributable to the fact that the magnitude ratio of the filter is 1.4 at that frequency. The system is clearly having difficulty tracking at 1.5 Hz, and the target is lost from the field of view of the cameras at higher frequencies.

A method of tracking higher frequency predictable motions without overshoot or

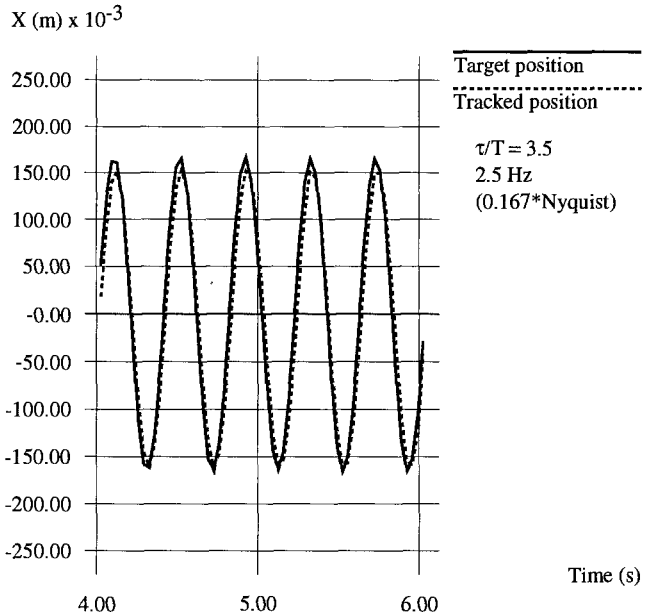


Fig. 6. Tracking performance of correlation prediction.

latency based on autocorrelation of the target signal is discussed in [3]. This method has been used to track sinusoidal target motions of 2.5 Hz and more. When the frequency of the target motion is held constant, the tracking errors with this approach are very small, even at high frequencies. An example of the tracking performance using this approach is shown in Figure 6. The correlation prediction technique makes no assumptions of the structure of the target motion signal—it should be effective for arbitrary periodic motions. However, the frequency content and amplitude of the target motion must change relatively slowly over time.

We have also combined a conventional predictive filter with the periodic motion prediction, such that both random motions with low frequency content and higher-frequency predictable motions can be tracked successfully with automatic switching between the two techniques. This has worked reasonably well, although transitioning between the two techniques successfully requires gradual changes in the target motion trajectory.

Numerous other algorithms for visual tracking have also been implemented using TRICLOPS, and are discussed in [4]. A quantitative measure of tracking performance, the tracking error bandwidth, is defined in [4] as the frequency of sinusoidal object motion at which the tracking error exceeds 20% of the object motion amplitude:

$$\frac{|i(j\omega)|}{f|\psi(j\omega)|} = 0.20. \tag{1}$$

where f = focal length, $i(j\omega)$ = image coordinates of target, $\psi(j\omega)$ = target position angle.

Using the above definition, an experimental tracking error bandwidth of approximately 1.1 Hz was obtained with a predictive filter. Without prediction, the tracking error bandwidth is about 0.34 Hz.

4 Model-Based Tracking

Of course, the tracking of simple targets in simple scenes, while useful to demonstrate the maximum performance (in terms of speed) which can be achieved, has limited application potential. A more useful capability is to be able to track an object of interest in a natural visual environment. At NIST, work toward developing such robust tracking has focused on model-based tracking of known target objects. Here, established geometric relationships between object features are used to improve the robustness of tracking. The basic elements of this approach are outlined below; additional information may be found in [5]. The approach described below assumes a rigid 3-D object with known geometry and appearance is being used as a target, and that it moves with 3 translational degrees of freedom.

The successive stages of computation for this tracking approach are:

1. Edge extraction - Extract the position and orientation for all edge points.

2. Data association - Determine likely groupings of edge points to each model feature.

3. Feature measurement - Use the grouped edge points to determine location of each feature.

4. Feature aggregation - Determine the overall location of the target by fitting the target model to the conglomerate of computed feature locations. The variance in each feature measurement is taken into account in this step to obtain the best linear unbiased estimate (BLUE) of target location.

5. Motion model update - Use the computed object location to update parameters (e.g. velocity, acceleration) that describe how the target moves as a function of time. This combination of the target location estimates over time weights each observation by the inverse of its covariance.

6a. Prediction of target location in next image - Extrapolate the motion model to predict target location in next image. Use predicted target location to determine corresponding predicted feature locations.

6b. Prediction of target location at next servo cycle - Predict the target location such that the motions of the head will keep the image of object centered in the field of view.

This strategy has been used to demonstrate 3-D target tracking using TRICLOPS 5., and also for vehicle following 6.. In the tests using TRICLOPS, the system was able to track and servo on the target as it moved at velocities of up to 1 m/s at a distance of 0.85 m. The equivalent angular velocity for this tracking speed is about 1.2 rad/s (69 deg/s).

The algorithm has demonstrated promising results in tracking the target in the presence of partial occlusion. The algorithm also was not confused by any of the background scenery it encountered. An example of a challenging scene during tracking is shown in Figure 7, with the corresponding edge image shown in Figure 8. The algorithm is also fairly robust to deviations from the assumed fixed orientation of the tar-

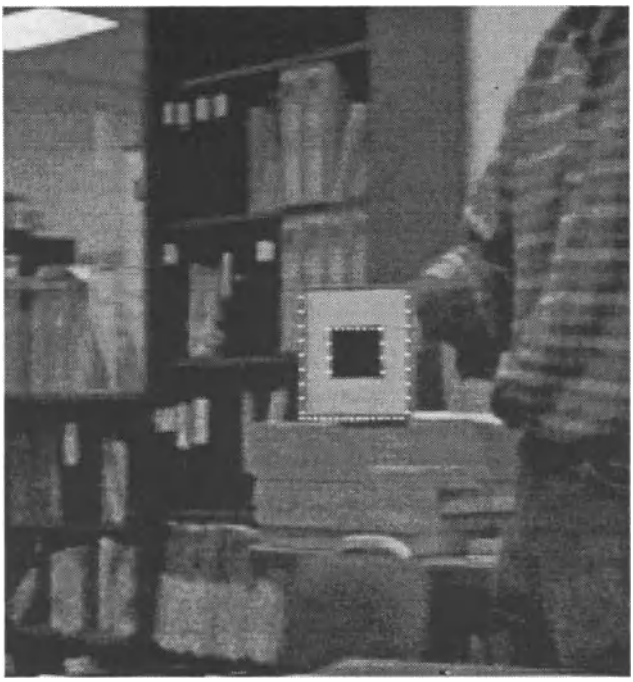


Fig. 7. Target with bookshelf in background.

Fig. 8. Edge image of target with bookshelf in background

get. The algorithm begins to fail when there are errors greater than 15 degrees in orientation. The algorithm will also fail if the target is moved with very sudden motions.

Progress has been made to extend this algorithm to tracking objects moving with a full 6 degrees of freedom of motion. Currently, however, the 6 degree of freedom version is not as fast, or as robust, as the 3-D version.

5 Conclusion

There are many aspects to defining the performance of visual tracking devices and algorithms. The performance is perhaps best defined by the restrictions which must be placed on the target object(s) and visual environment in order to achieve successful tracking. Examples of such restrictions include the maximum speed and frequency content of target motion, the maximum number of targets which may be tracked, lighting and background constraints required, the amount of a priori knowledge of the target object required, and allowable variability in the types of objects which may be tracked.

We have implemented a class of visual tracking algorithms using TRICLOPS which have concentrated on optimizing the performance criteria of speed and target maneuverability (at the expense of reduced target and image complexity). Using this approach, very high tracking speeds have been obtained. We have also implemented model-based tracking algorithms which lift the constraints on image complexity and work for an entire class of prismatic objects with known geometry (with some reduction in the allowable target speed). In both cases, the primary factor affecting the speed at which targets may move and be tracked successfully is the amount of image processing delay, and the effectiveness of the prediction used to compensate for it.

References

1. Fiala, J. C., Lumia, R., Roberts, K. J., Wavering, A. J., "TRICLOPS: A Tool for Studying Active Vision," *International Journal of Computer Vision*, Vol. 12, no. 2-3, pp. 231-50, April, 1994.
2. Albus, J. S., McCain, H. G., Lumia, R, "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," NIST Technical Note 1235, 1987.
3. Wavering, A. J., Lumia, R., "Predictive Visual Tracking," *Proc. SPIE Intelligent Robots and Computer Vision XII*, Boston, MA, September, 1993.
4. Fiala, J. C., Wavering, A. J., "On Visual Pursuit Systems," NIST Internal Report 5513, October, 1994.
5. Schneiderman, H., Wavering, A. J., Nashman, M., Lumia, R., "Real-Time Model-Based Visual Tracking," *Intelligent Robotic Systems '94*, Grenoble, France, July, 1994.
6. Schneiderman, H., Nashman, M., Wavering, A. J., Lumia, R., "Vision-based Robotic Convoy Driving," to appear in *Journal of Machine Vision and Applications*.

Developing an Active Observer

Jan-Olof Eklundh, Tomas Uhlin, Peter Nordlund, and Atsuto Maki

Computational Vision and Active Perception Laboratory (CVAP)*
Department of Numerical Analysis and Computing Science
KTH (Royal Institute of Technology), S-100 44 Stockholm, Sweden
Email: joe@bion.kth.se

Abstract. Seeing robots are usually aimed at functioning in real environments. Hence the figure-ground problem is essential. We argue that for a seeing robot, capable of actively fixating and holding gaze on objects in three dimensions, this problem is manageable, in particular if multiple cues can be used. An important point here is that a seeing robot should be able to utilize what the environment offers, rather than relying on a predetermined set of features.

Furthermore, if there are early processes for figure-ground segmentation the local statistics of an observed (yet unknown) objects should be simpler than that of the entire scene. That suggest that processing to derive further properties, of e.g the shape or motion of that object, or even recognizing, could be based on other features than those used to segment out the object. Such a view fits well with recent theories about view-based recognition. It also allows efficient implementations in terms of a visual-front-end, since both the target selection and the target analysis can be based on the output from such a layer, even though different features are used.

1 Introduction

Vision as we know it from biology is an active process. The seeing agent uses vision as well as its other senses to guide its behaviors. Current research in active computer vision studies a range of such systems, such as the prey catching and threat avoiding system of Arbib and Liaw [1] that uses very simple perception, and systems such as the one presented by Kosecka *et al.* [4], that involves rather complex perceptual processes.

A major emphasis in active vision research has not surprisingly been on problems similar to those traditionally studied in computer vision. Typical efforts deal with navigation, manipulation and recognition tasks in indoor and outdoor environments usually encountered by man. Such problems have for a long time been addressed without reference to vision as an active process. However,

* The authors want to thank Kourosh Pahlavan for valuable discussions on the topic of the article. The reported work has been supported by a grant from the Swedish National Board for Technical and Industrial Development. This support is gratefully acknowledged.

despite considerable progress on computational techniques for deriving information from the scenes from images of it, and despite striking successes in cases such as vision-based vehicle guidance, one can hardly claim that "seeing robots" capable of advanced and flexible behavior in rich environments exist. This is especially obvious when human performance is used as the (implicit) yardstick for measuring success, which is often the case.

One particular and ubiquitous problem that in the authors' view has prevented the application of state-of-the-art and theoretically well-founded techniques of traditional computer vision to real-life situations is what is generally called the figure-ground problem. A major reason for this is that each such technique makes very precise assumptions of what features are needed, while consistent extraction of these features often turn out to be very difficult in complex environments. In fact, they may not even be available. On the other hand we know that humans most of the time experience few problems in segmenting out objects in the surrounding world. Of course, these objects are of many types and sizes, some complex and composite, while others are simple and only parts of larger entities. As we know, what constitutes an object also depends on the task at hand and the context. Nevertheless, humans have few problems with this, while bootstrapping computer vision algorithms for the analysis of the same type of scenes seems extremely difficult.

In this paper we argue that such problems have viable solutions in active vision, which assumes a system that acquires visual information by looking at a *scene* rather than at images of it - it goes beyond pictorial vision. We suggest that this leads to an approach considering the figure-ground problem in the three dimensional world, and separating it from the later stages of the analysis. Well-known techniques are still used, but by adapting to what the world offers we will in fact facilitate their applicability by limiting the context. Our arguments are based on several observations that we will elaborate on in subsequent sections. We will then present some experimental evidence supporting our view and discuss their consequences.

1.1 The active observer and the figure-ground problem

Consider a person moving around in the world, while looking at various locations and things in his way. Objects will then due to his own or their motion enter and leave his field of view. He will sometimes shift his gaze to such objects to find out, e.g. if their trajectory crosses his path, or, to determine what kind of things they are. In fact, he may be looking for specific types or instances of objects, for example they may be obstacles to avoid, or something he needs.

Certain capabilities that this observer displays are of importance for a seeing robot as well. First, the observer is capable of singling out things that stand out from their immediate surroundings. Secondly, he can hold these things in his field of view long enough to identify and derive various properties of them.

The first problem is the mentioned figure-ground problem, but it can be noted that often something is considered as an object just because it stands out as a three-dimensional entity. Hence, cues indicating that are essential. This is

not surprising, since camouflage in principle is impossible in three dimensions. It is well-known that motion, binocular disparities, and blur give such information with greater certainty than monocular image cues. Hence, early use of three dimensional information seems important, as is also suggested in the study of human attention by Nakayama and his co-workers, see e.g Nakayama and Silverman [6] and Shimojo *et al.* [9].

In active vision such information is readily available. Binocular head-eye systems capable of fixation can, as was pointed out early by Ballard [2], observe three-dimensional structure in the world in relation to the fixation point. By shifting fixation this can be applied to various locations in the scene. That effect is present both when binocular disparities and motion parallax is considered. As discussed in Pahlavan *et al.* [8] blur as a depth cue is conveniently available to an active system capable of controlling its optical parameters.

When a target is detected, an active system can hold gaze on it. This can be done by image tracking, but pursuit movements are needed if the target tends to move out of the field of view. There are several important consequences of such a capability. First, the earlier attention step has created a region of interest that corresponds to an actual physical target (if that step has accomplished its task). The characteristic features of that target can hence be found in this region. It is reasonable to assume that the statistics of various features are simpler there than over the scene as a whole. Furthermore, if the target is masked out, then any typical feature of the object can be used to characterize or recognize it, whether this feature is common in the background or not. This becomes particularly important in a dynamic situation, when the background will vary independently of the object. Since the target is seen for some time, one can also assume that certain characteristics can be ascribed to it. Note that the target has an identity before one knows what it is. These characteristics can then be used to compute and maintain information about e.g. its shape and motion, but also to establish that it is the same object when it stops or changes motion pattern, or when only a fraction of it is seen due to occlusion: The target that's singled out forms its own model. Moreover, what is used to analyze it further, is in principle independent of what was used to identify it as being an object.

To effectively apply a technique like the one we've outlined a third required capability becomes obvious: Many different cues must be computed concurrently, and the system must be able to choose those that at a particular instance of time are available and distinctive. This suggests a *systems approach* to the problem, in which many different processes are integrated and controlled over time. It also suggests that computations made on the image data are reused whenever possible, for efficiency. Recent research on Visual-Front-End models for early vision, see e.g. Gårding and Lindeberg [3], provides possible model for that.

In this paper we'll present experiments that show that this is a promising computational framework in which active vision plays an important role. In principle, the information needed to implement a scheme like this can be derived by traditional methods. Here we will try to show that active vision is especially suited for it.

2 The System Description

The principles described above have been implemented in a system embodying an autonomous mobile observer. This observer should while moving be able to detect and smoothly pursue moving or stationary targets binocularly while maintaining vergence. Currently, motion and binocular disparities are used in the figure-ground separation step, but the architecture is open for additional cues. So far fundamental skills in terms of fixation, target pursuit and target discrimination, have been implemented. The basic system partly runs in real-time on an existing mobile platform, and partly as a post-processing stage working on images captured in real-time during the execution of the former processes.

Mechanisms for discovering moving targets provide means of shifting attention. In other words, our system has two components, one to maintain attention, and another one to find and select new locations to attend to. Moreover, these function in parallel, because the system would otherwise be without choices. They are implemented in the form of a pursuit and a motion detection mechanism, thus obtaining what we believe is the most basic behavioral level for an active observer (Figure 1). (Currently, motion provides the only cue to changing attention, but other cues can easily be incorporated.)

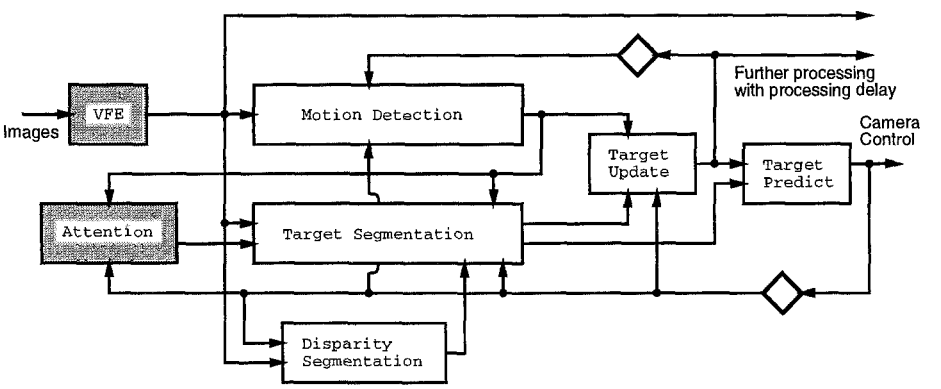


Fig. 1.: The system implementation is shown schematically. (The diamonds indicate a one frame delay.)

A note on terminology. The term *target* will denote the portion in the image which is currently attended and supposed to represent a scene-object. A *target model* is used, which is coded as an image mask, including pixels that are considered as belonging to the target. All segmentation modules produce similar masks after a thresholding step. These masks are used for the integration.

2.1 Attention and Smooth Pursuit

A key feature of the system lies in its ability to smoothly pursue an arbitrary target that is described by its location, extent and visual appearance. No a priori knowledge about the target's visual appearance, i.e. texture or shape,

is built into the system, although additional constraints or knowledge can be included dynamically as information becomes available, and similarly that such information can be excluded as it becomes obsolete. The features that are seen, are assumed by the pursuit module to change smoothly. Non smooth changes will trigger attentional mechanisms.

The implementation is based on a coarse to fine correlation scheme similar to the one reported in Pahlavan *et al.* [8]. This technique works well when there are no occlusions. We will introduce an extension handling occlusions as well.

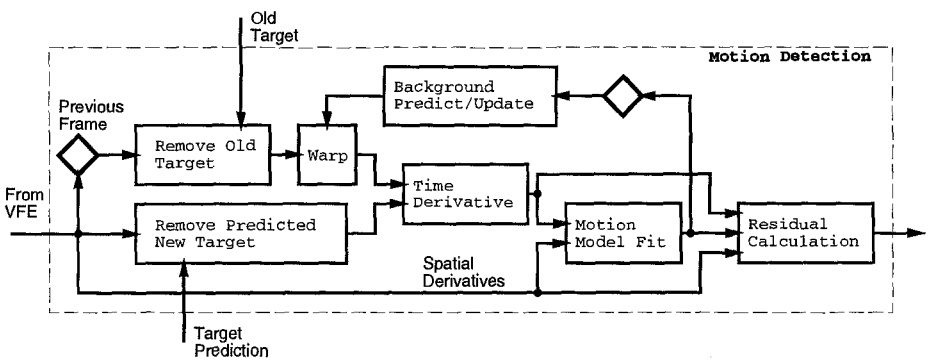


Fig. 2.: The motion detection is shown schematically. This is a detailed description of the `Motion Detection` box in Figure 1. (The diamonds indicate a one frame delay.)

To handle occluding objects and distracting things in the background, motion detection is integrated to filter out parts of the scene that can be parts of the target, namely those that are moving. This is provided in the system by a connection from `Motion Detection` to `Target Segmentation`, see Figure 1. We will see later in the experiments that this is not enough in many cases when there are occluding objects that are themselves moving, since they will also be detected as moving. Binocular disparity can in these situations aid in depth discrimination by also providing a clue to where occluding objects may lie.

Motion Detection Motion detection allows the system to shift its attention to another location, if the system "looses interest" or fails to pursue the current target. Control mechanisms of the former type are not present in the current system, but could easily be included. Motion detection functions continuously during pursuit and is used in selecting points where the target is lying (Figure 2).

Using the brightness constancy constraint a global affine image velocity model is fit to the gradient and time derivative information (see `Motion Model Fit` in Figure 2). Two steps involving feedback are included to account for object motion and large background motion.

- The predicted and previous position and extent of the target are used to mask out parts of the image which likely belong to the object, so that they do not affect the calculation of the affine parameters for the background. See

feedback into `Motion Detection` in Figures 1 and 2. This is a crucial step to avoid iterations that otherwise would have been necessary to get reasonable performance even if targets are large. We could say that the iterations are performed in time instead of in a single frame.

- The accumulated affine parameters are used over time to cancel out the majority of the time difference, see feedback into `Warp` in Figure 2. Thereby we can avoid iterations in fitting the affine model, and instead iteratively refine our motion model in time. This is increases the computational speed and decreases the subsequent lag of the pursuit mechanism. We have here an example of how one can tradeoff the performance of a single algorithm for better overall performance.

Once the fit is made to the background, and the affine parameters are available, the residual normal image velocity is calculated in the entire image, `Residual Calculation` in Figure 2. The obtained residual is thresholded and high residuals are considered as possible independently moving areas in the scene. The threshold on the residual is adaptive and is set relative to the difference between target and background motion.

Target Segmentation The aim of target segmentation is to determine which parts of the scene are moving consistently with what is currently believed to be the target. This is the typical situation in connection with occlusions, when the target gradually disappears, to later emerge on the other side of the occlusion. It also supplies the system with the ability to discriminate between several moving targets, found by the motion detection, and distinguish parts of the scene that lie at the same depth, found by the disparity segmentation (Figure 3).

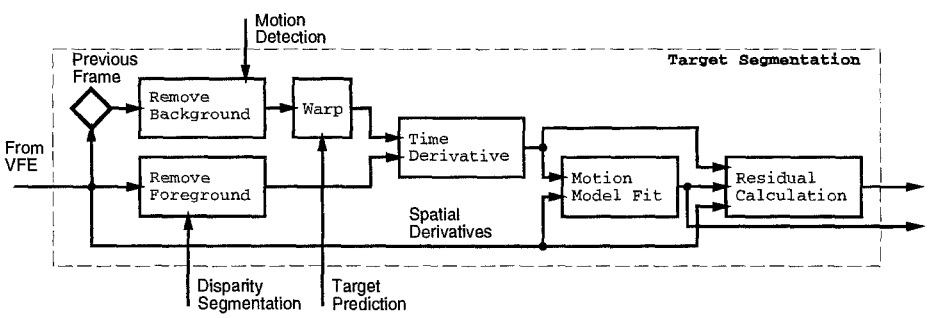


Fig. 3.: The target segmentation is shown schematically. This is a description of the `Target Segmentation` box in Figure 1. (The diamonds indicate a one frame delay.)

The calculations on the target are performed in analogy with what is done for the background motion, with the advantage that most of the computations spent on the background can be used also here (spatial derivatives and smoothed images from the VFE). By sharing these data we greatly increase the performance.

Analogous to the motion detection in Section 2.1, the residual normal image velocity is calculated and an adaptive thresholding is applied.

Disparity Segmentation The system in addition computes relative depth information. Having a disparity measure is also necessary to obtain vergence.

The disparity computation is carried out with a phase based method which fits well into the VFE framework since the same data can be used for vergence and pursuit movements. It also performs well in a real-time implementation for vergence on the KTH-Head-Eye system, see Uhlin *et al.* [10].

The object of disparity selection is to select the disparities that belong to the target in the presence of disparities that arise from other locations in the scene. The method is described in Maki *et al.* [5] and has been further developed to take advantage of the feedback obtained when the system is run in a closed loop together with other cues to follow a single target, to ensure that version and vergence components in smooth pursuit remain consistent. The basis for selection is the predicted location and extent of the target.

2.2 Integration

The integration is performed mainly in `Target Update` (Figure 1), but in the `Target Segmentation` (Figures 1,3), motion detection information and disparities are also integrated. The target extent is kept in a mask. The integration produces an updated target mask using masks produced in the paragraphs "Motion Detection" and "Target Segmentation" in Section 2.1. There are two parts needed to update the target mask, one to exclude pixels and and one to include pixels.

Target areas that do not get support from either motion detection or target segmentation are excluded from the target model. Also the disparity module detects areas in the scene that lie in front of the pursued target, which are then excluded from the target model.

Areas in the scene that are detected as both moving independently, detected by motion detection, and are moving consistently with the target image velocity model from the target segmentation, are added to the target model.

3 Real-Time Implementation

Presently we have implemented parts of what has been described above to perform real-time visual processing coupled to motor control of the KTH Head-Eye system. Hence, there exist mechanisms for pursuit, motion detection and disparity estimation running at 25 Hz. See Uhlin *et al.* [10] for more details.

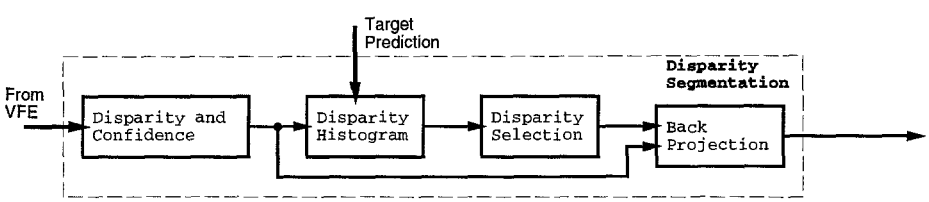


Fig. 4.: The disparity segmentation is shown schematically. This is a detailed description of the `Disparity Segmentation` box in Figure 1.

4 Experiments and Results

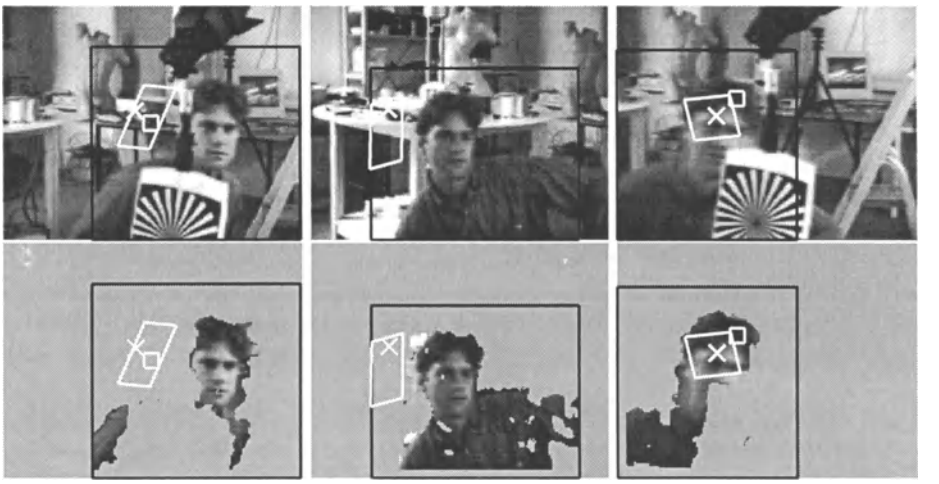


Fig. 5.: Resulting target pixels as extracted by the system (after integration and update) from the sequence shown in the top row. Using motion detection and an affine image velocity model for the target we can find areas that belong to the target such that occluding objects are removed and a correct pursuit is performed although we have partial occlusion. The pursued target is marked by the small rectangle. The small rectangle marks a fix point on the background as computed by the optokinetic mechanism. The pursuit is performed at 25 Hz, and shown are every 60th frame.

We will briefly indicate how the system performs in a few different situations, both in the presence of all cues and when some cues are not present. The target in most examples undergo complex movement, including rotations and movements towards and from the camera, while the observer undergoes ego-motion. No knowledge about the motion is assumed. Additional examples are presented in Uhlin *et al.* [10], where also more details are given.

The first example, Figure 5, shows a laterally moving target, a human, the derived masks, a fixed point in the background and the estimated local motion. That example contains only one moving target, while Figures 6 and 7 demonstrate the multiple target case. By using disparities together with motion the system can maintain the identity of the observer target also when other targets appear. Figure 8, finally, shows how properties of the target, here simple luminance features, indeed can be determined when it is singled out.

5 Discussion

We have argued that an active observer by dynamically fixating in the three dimensional world can single out objects of interest. This suggests an approach

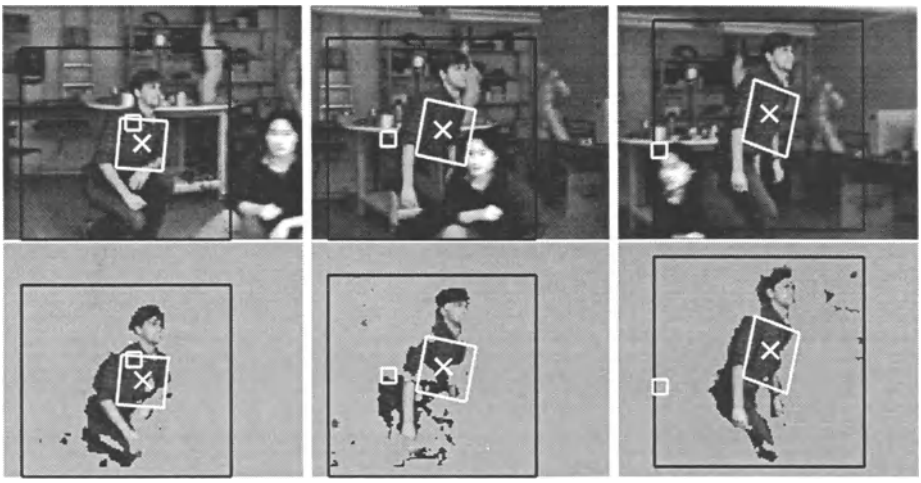


Fig. 6.: A sequence captured during pursuit. The pursuit is performed at 25 Hz, and shown are every 18th frame.

Fig. 7.: Target pixels as extracted by the system from the sequence shown in Figure 6, but without the disparity cue. The attention shifts to the other moving person. Every 18th frame is shown here.

in which the figure-ground problem and subsequent processing in principle is independent, although based on similar information. It also suggests that known methods for e.g. feature extraction and recognition may be facilitated, since local statistics is simpler.

References

1. Arbib, M.A. and Liaw J-S. (1995). Sensorimotor Transformations in the World of Frogs and Robots. Journal of Artificial Intelligence. **72**, 53–79.
2. Ballard, D.H. (1991). Animate vision. Journal of Artificial Intelligence. **48**, 57–86.
3. Gårding, J. and Lindeberg, T. (1995). Direct Computation of Shape cues Based on Scale-Adapted Spatial Derivative Operators. International Journal of Computer Vision. To appear.
4. Kosecka, J, Christensen, H. I., and Bajcsy, R. (1995). Discrete Event Modelling of Visually Guided Behaviors. International Journal of Computer Vision, **14**, 179–191.

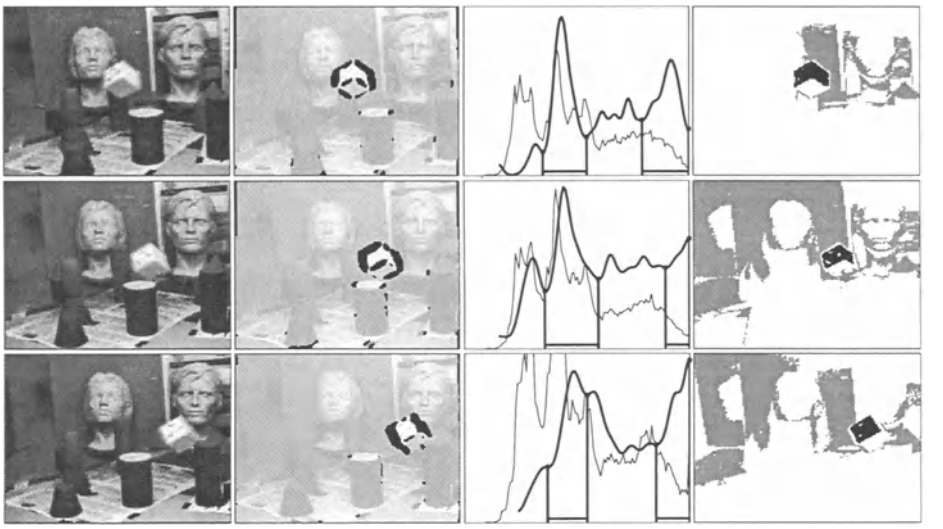


Fig. 8.: 1st column shows original images. 2nd column shows motion mask obtained with simple motion-algorithm. 3rd column shows grayscale histogram. The histogram for the original image is indicated with a thin line. The histogram for just the area in the motion mask is indicated with a thick line. Straight lines indicate the ranges used to threshold out the object, respectively the background. Compare the significant difference between the two histograms. The 4th column shows the thresholding results. Object marked in black, background marked in gray. Only the connected regions overlapping the motion mask most in respective thresholding range are shown.

5. Maki, A., Uhlin, T., and Eklundh, J.-O. (1994). Disparity selection in binocular pursuit. In *Proc. 4th IAPR (International Association for Pattern Recognition) Workshop on Machine Vision Applications*, pp. 182–185.
6. Nakayama, K. and Silverman, G. H. (1988). Serial and parallel processing of visual feature conjunctions. *Nature*, , no. 320, 264–265.
7. Pahlavan, K. Uhlin, T. and Eklundh, J-O. (1993a). Active vision as Methodology. Active Perception. Aloimonos, Y. (ed.): Lawrence Erlbaum Associates, Hillsdale, NJ.
8. Pahlavan, K., Uhlin, T., and Eklundh, J.-O. (1993b). Dynamic fixation. In *Proc. 4th International Conference on Computer Vision*, pp. 412–419, Berlin, Germany. IEEE Computer Society Press.
9. Shimojo, S., Silverman, G. H., and Nakayama, K. (1988). An occlusion-related mechanism of depth perception based on motion and interocular sequence. *Nature*, , no. 222, 265–268.
10. Uhlin T., Nordlund P., Maki A., and Eklundh J.-O. (1995). Towards an active visual observer. Tech. Rep. ISRN KTH/NA/P--95/08--SE, Dept. of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology). Shortened version in Proc. 5th International Conference on Computer Vision pp 679–686.

Active Stereo Vision System with Foveated Wide Angle Lenses

Yasuo Kuniyoshi* Nobuyuki Kita* Sebastien Rougeaux*
Takashi Suehiro**

* Autonomous Systems Section,
Intelligent Systems Division,
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba,
Ibaraki 305, JAPAN and

** Tsukuba Research Center
Real World Computing Partnership
Tsukuba Mitsui Bldg. 17F,
1-6-1 Takezono, Tsukuba, Ibaraki 305,
JAPAN

Abstract. A novel active stereo vision system with a pair of foveated wide angle lenses is presented. The projection curve of the lens is designed so that it facilitates active vision algorithms for motion analysis, object identification, and precise fixation. A pair of such lenses are mounted on a specially designed active stereo vision platform. It is compact and light so that it can be mounted on a mobile robot or a manipulator. A real time stereo tracking system is constructed using the platform, a dual-processor servo controller, and pipelined image processors with a multi-processor backend.

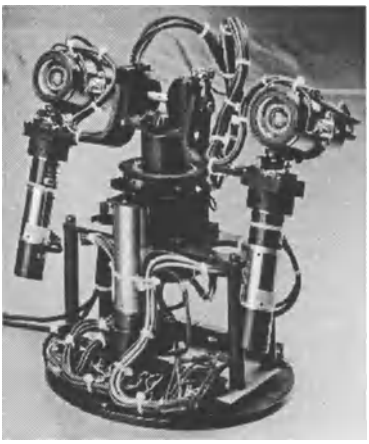
1 Introduction

For purposive robots in a complex and dynamic world, the most basic visual functions required are target detection, fixation, and tracking. A wide field of view is necessary for finding an important target around the robot, whereas a high resolution is desired for close observation or stereo fixation on the target.

In this paper, we first present our active stereo gaze platform *ESCHeR*(E[TL] S[tereo] C[ompact] He[ad for] R[obot Vision]). It is compact and light weight so that it can be mounted on a mobile robot or a manipulator. The most distinguishing feature of *ESCHeR* is that it is equipped with foveated wide angle lenses. The design and implementation of the lenses is described in detail. It is strongly motivated towards applications for active vision in complex and dynamic environments. Finally, an integrated real time stereo tracking system is presented. It consists of the gaze platform, a dual-processor servo controller, and pipelined image processors with a multi-processor backend, interconnected via fast communication links.

2 *ESCHeR:* ETL Stereo Compact Head for Robot Vision

Figure 1 shows our active stereo gaze platform. It has the following mechanical structure:



Dimensions			
Width	222mm	Height	187mm
Depth	160mm	Baseline	180 mm
Weight	2.0 kg		

Performance				
Axis	Range	Max. Vel.	Max. Acc.	Resolution
Pan	200°	$140^{os^{-1}}$	$04000^{os^{-2}}$	0.0044°
Tilt	090°	$350^{os^{-1}}$	$14000^{os^{-2}}$	0.0145°
Left	100°	$400^{os^{-1}}$	$16000^{os^{-2}}$	0.0125°
Right	100°	$400^{os^{-1}}$	$16000^{os^{-2}}$	0.0125°

Fig. 1. *ESCHeR:* ETL Stereo Compact Head for Robot Vision

- It has four DOFs for gaze control: Independent vergence for each cameras and a common tilt supported by a common pan. Each is driven by DC motors mounted with encoders.
- Two CCD color cameras equipped with foveated wide angle lenses. Each lens has a stepper motor for computer controlled focus. The optical properties of the lens are discussed later.

As shown in Fig. 1, the general mechanical performances of our platform can be compared to Sharkey et al.'s [1] system. Furthermore, achieving small size and light weight, it can be mounted on a mobile robot and used for active navigation.

3 The Foveated Wide Angle Lens

Previous approaches to obtain space variant images include: (1) Using a zoom lens to acquire variable resolution frames sequentially [2]. (2) Using a pair of wide/tele cameras [3]. (3) Converting from normal images by digital warping [4, 5]. (4) Using a space variant CCD chip [6].

These methods have various problems such as co-axiality and continuity of the images, time delays, efficiency, and resolution.

Very recently, an alternative approach has been proposed by Suematsu et al. [7] – A special lens with a space variant optical projection. Their goal was to mimic the human visual acuity curve as closely as possible. But there remains important questions: What is the advantage of such a lens? How can we utilize it? Is there any better design?

We present a novel design of a space variant lens which facilitates useful vision algorithms and achieves higher resolution variance (e.g. tele/wide range).

Our approach avoids the problems with other methods. Also, it considers the optical projection process which cannot be handled by pure 2D methods like warping or special CCDs. This becomes critical when we want to achieve a wide view angle. Even if we use the space variant CCD, we must use some wide angle lens, which generally deviates substantially from the standard (non-distorted) projection. Therefore, if we ignore the optical process, we cannot obtain a desired image. Integrating the optical process and the 2D image transformation is very important in achieving a foveated wide angle vision.

3.1 Design of the Lens

Our motivation to build a useful active vision system for purposive robotic behavior leads to the following design principles.

1. The lens should have a super wide view angle and a high resolution fovea. The resolution range should be as wide as possible.
2. The projection curve should be meaningful in terms of vision processing, especially motion analysis, stereo fixation/tracking, quick pattern identification and detecting a coarse structure of the environment.
3. The mechanical characteristic of the lens should be suitable for active camera movement such as saccade and pursuit. This requires small size and light weight, computer controllable optical DOF, etc.

The core part of the lens design is to specify a *projection curve*, which is a mapping from every incident angle of optic rays entering the lens system to a corresponding image height on the CCD surface, both measured from the optical axis.

The curve should be such that it has a high magnification at small incident angles and a low magnification and a high distortion (condensing the image size) at the periphery. We took the following design procudure: (1) Define two projection curves adequate for the fovea and the periphery. (2) Find a useful projection curve which smoothly connects the fovea and the periphery. (3) Solve the curve equations to find appropriate parameters for an integrated projection curve.

The above procedure is important because it allows us to assure that each part of the projection curve is grounded to useful vision algorithms.

Fovea curve: The primary functions of the fovea are to precisely extract local image features (e.g. for stereo matching or shape recognition). The standard projection (as in normal cameras) is adequate. Most current vision algorithms assume this projection, so there is no limit in application. A human fovea has a more complicated structure with nested spatial variance, e.g. a variable resolution curve within a fovea. But we decided that the standard projection is more useful from the application point of view.

Periphery curve: The peripheral area mainly copes with target detection, self motion analysis and extracting a global structure of the environment. Also, optics tells us that we must allow a high distortion if we wish a super wide view angle. We selected a spherical projection for the periphery.

There are at least two strong vision algorithms which support this curve: 1) Nelson and Aloimonos[8] showed that a spherical projection greatly simplifies an optical flow analysis for separating a target motion from the ego-motion. Also it contributes to the robustness of the solution. 2) Morita et al.[9] showed that under a spherical projection, motion stereo for long straight lines can be simply done by detecting a common intersection of them. This can be used in detecting a global structure of the environment while a robot is moving around.

The fact that the human optical system adopts the spherical projection also motivates us toward the selection of the curve [10].

Transition curve: In this step we must find a function to connect the fovea and the periphery curves in a meaningful way. Since the slope of the fovea curve is steep and that of the periphery is blunt, the transition curve must have a monotonically decreasing first order derivative.

A logarithmic function is one candidate. As first introduced to the computer vision community by Sandini [6], the human visual system adopts a log-polar mapping between the retina and the visual cortex. They have shown various advantages of this mapping in terms of concrete vision algorithms[11]: It enhances the lock-on effect in stereo fixation, simplifies the optical flow analysis, and it introduces image invariance to rotation and scaling which is quite powerful for 2D shape identification.

We found that the logarithmic of the spherical projection can connect both the fovea and the periphery smoothly. If we apply a digital warping technique to the acquired image and transform it from the orthogonal (CCD) to the polar coordinates, we achieve a log-polar mapping. Thereby we can assign a concrete meaning to the transition curve function.

Theoretical projection curve: Following the above discussion, we define a projection curve using the following equations (1–3), which relate image height y to incident angle θ. f_i denotes the focal length of the i'th part of the curve.

Equation (1) is a standard projection. Equation (3) is a "shifted" spherical projection. The shifting term q is introduced in order to assure a connect point. Equation (2) is a logarithmic of the spherical projection. During the image processing, the shifting term can be canceled by simply subtracting the offset from the radial coordinate of each image point.

$$y = f_1 tan(\theta), \quad 0 \leq \theta \leq \theta_1 \tag{1}$$

$$y = log_a(f_2\theta) - p, \quad \theta_1 \leq \theta \leq \theta_2 \tag{2}$$

$$y = f_3\theta + q, \quad \theta_2 \leq \theta \leq \theta_{max} \tag{3}$$

By solving continuity conditions on zeroth and first order derivatives, we arrive at the following relations between the parameters.

$$f_2 = f_1 \tag{4}$$

$$f_3 = (f_1\theta_1 \sec^2\theta_1)/\theta_2 \tag{5}$$
$$a = exp(\cos^2\theta_1/(f_1\theta_1)) \tag{6}$$
$$p = f_1(\theta_1(\log f_1 \sec^2\theta_1 + \log\theta_1 \sec^2\theta_1) - \tan\theta_1) \tag{7}$$
$$q = f_1(\theta_1(\log\theta_2 \sec^2\theta_1 - \log\theta_1 \sec^2\theta_1 - \sec^2\theta_1) + \tan\theta_1) \tag{8}$$

It should be noted that the curve (eq.1–3) simulates the overall image transformation in the human visual system, e.g. a combination of a spherical optical projection and a log-polar retino-cortical mapping [10]. The retino-cortical mapping is confirmed for the visual field less than 15 [deg] (half angle) from the fovea [6] (eq.2). To our knowledge, no counter fact has been found against the hypothesis that the peripheral area beyond 15 [deg] adopts a uniform retino-cortical mapping, which means the image transformation in this part is a pure spherical projection (eq.3).

3.2 Implementation of the Lens

Deciding parameters: Theoretically, a designer can choose and set an arbitrary subset of parameters and solve the above equations for the rest. But the real optical design imposes strong constraints on the parameter selection.

The first barrier in the optical design is to achieve both a high resolution fovea and a super wide view angle in a single lens system. Thus, we must first decide f_1, the focal length of the fovea, and the maximum view angle θ_{max}. They are decided through several case studies in optical design, and considering the limitation on size and weight. In order to achieve a small lens size, we decided to use a 1/3" CCD.

Next, the connect point parameters, θ_1, θ_2, are decided. They are mutually dependent and constrained by θ_{max}, f_1, max_x, but have one degree of freedom, so we adopted an iterative simulation to find a good compromise. Then, all the other parameters are solved by equations (5)–(8), as shown in Table 1.

Parameter	Value	Parameter	Value
Foveal resolution [deg/pixel]	0.05104	Maximum view angle θ_{max} [deg]	60.02
[mm/pixel] at 1[m]	0.8908	Horizontal CCD size max_x [mm]	2.44
Peripheral resolution [deg/pixel]	0.3942	Foveal/middle focal length	
[mm/pixel] at 1[m]	6.881	$f_1 = f_2$[mm]	10.70
Connect point 1 (θ_1) [deg]	2.584	Peripheral focal length f_3[mm]	1.385
[pixel]	50.66	a	7.909
Connect point 2 (θ_2) [deg]	20.0	p [mm]	-0.8352
[pixel]	154.5	q [mm]	0.9889

Table 1. Curve parameters. (Angles measure half angles [deg.]. Angular resolution is measured in degree per pixel assuming a 512 × 512 pixel digital image.)

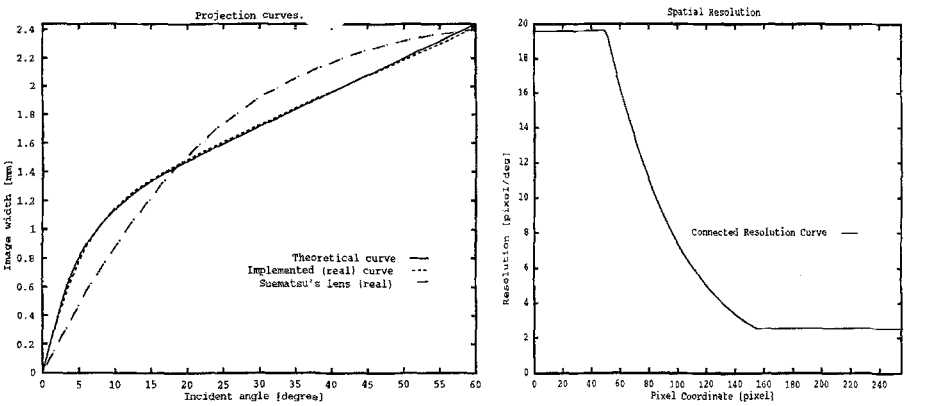


Fig. 2. Theoretical and the real (implemented) projection curves of our lens (left), compared with the real projection curve of the Suematsu's lens[7] (Data by courtesy of Prof.Suematsu). Spatial resolution curve (right) : Pixel coordinate – resolution [pixel/deg].

Figure 2(left) shows the theoretical projection curve for the decided parameters. The spatial resolution curve, Fig. 2(right), is obtained by differentiating the inverse projection function by image height and taking its inverse.

Optical design: The optical design was conducted as follows: First, achieve the fovea focal length while keeping the wide field of view. Then, adjust the projection curve to the theoretical curve while eliminating aberrations as much as possible. Each step is done by an iterative raytracing (from incident to image) and fitting process.

Focal length (fovea)	10.7[mm]	No. of aspherical surfaces	2
Focal length (60[deg])	1.39 [mm]	Max. image height	2.42[mm]
View angle (2θ)	120[deg]	Total optical length	60[mm]
F No. (fixed)	4.0	Optimal target distance	1.5[m]
Configuration	10 groups 11 lenses	Resolution (fovea)	30 [lines/mm]

Table 2. Final specification of the implemented lens.

The final optical specification of our lens is shown in Table 2 and Fig. 3(left). In order to achieve extremely high distortion required for the first step, we placed a concave surface toward the object as the first surface, and an opposite surface at the last surface[1]. This pair contributes most to the distortion control. Since

[1] There are two shortcomings in the use of concave first surface. It imposes a limitation on the field of view (120[deg]) and promotes the peripheral brightness drop. But if we use a convex first surface, it was impossible to achieve the desired resolution variance.

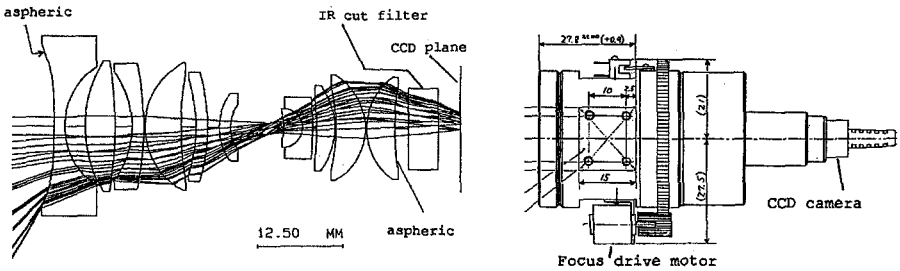


Fig. 3. Lens configuration overlayed with a ray tracing (left) and harness design (right).

we need to control the distortion independently of other aberrations, we decided to use aspherical surfaces for this pair.

Fig. 2(left) shows the real projection curve (based on the ray tracing) of our implemented lens. The real curve slightly deviates from the theoretical curve. Compared to the theoretical curve, the real curve starts to get blunt a little bit earlier and the transition zone is a little bit wider (in other words, with a slightly larger curvature). The maximum deviation is 35.7 [μm] at θ=3.5[deg]. Dividing it by a digitized (512×512) pixel size, 9.53[μm], the deviation is 3.75[pixels]. Other zones less deviate from the theoretical curve. The deviation is around 10 [μm] or less. According to the inferrometer measurement of the real lens surfaces, they are fabricated precisely to the optical design such that the curve deviation is much less than one pixel throughout the entire field of view.

For comparison, the projection curve by Suematsu et al[7] is overlayed on ours Fig. 2(left). The difference in the slopes of the foveal range shows that our lens has a stronger visual acuity. Also, our lens has a constant low image compression rate in the periphery compared to Suematsu's, which is useful for periphery-based object/motion detection.

Figure 4 shows a real image taken from the current version of our lens. It is observed that the brightness drops towards the periphery rather significantly, which can cause some trouble in image processing. This characteristic is intrinsic to any wide view lens[2]. We are currently working to reduce this brightness variance by a graded coating technique.

Harness design: Figure 3(right) shows the harness design of the lens. The size is 35.6[mm] diameter and 73.9[mm] length and the weight is 137[g] plus 5[g] focus motor[3].

[2] The optics tells us that the light intensity on the screen is typically proportional to $cos^4\theta$, where θ is the incident angle

[3] In order to reduce the size and weight so that the lens can be swung at high speed and acceleration, we had to make the following decisions: 1) Drop an iris mechanism. Based on the brightness calculation, we decided that the automatic gain control and controllable digital shutter equipped with the CCD camera controller can replace the mechanical iris. This decision reduced the total weight to 1/2. 2) Specific camera. In

Fig. 4. Test image from the real lens.

4 System Integration

The overall architecture of the active vision system is shown in figure 5.

Each joint of the gaze platform is driven by the *servo-controller*, a dual INMOS T805 transputer based architecture which receive asynchronous commands issued from vision processing and converts them into a smooth, 500 Hz interpolated series of commands for the motors.

The *image processing unit* is based on a DataCube MaxTD system, controlled by an MVME 167 CPU board running a LynxOS kernel. Stereo image acquisition is performed by a Digicolor using red and green components of the video signal. The real-time image processing is performed jointly with one or two MaxVideo 200 for pipeline operations (like thresholding, smoothing, blurring, ...) and a KiwiVision, a set of nine INMOS T805 transputers with a MaxBus interface for extracting features and interfacing to motor controllers via INMOS links (20Mbps).

5 Conclusions and Future Work

We have presented a design and implementation of a novel stereo tracking system with foveated wide angle lenses.

---

order to shorten the back focus length, we had to limit a valid camera model. We use a tiny 1/3" CCD camera ($\phi 12 \times$ L31.4 [mm]). If we adopted the standard C mount, the size would be around 3/2 of our current design. The focusing mechanism is driven by a tiny stepping motor and limit sensors. These can be interfaced to a host computer.

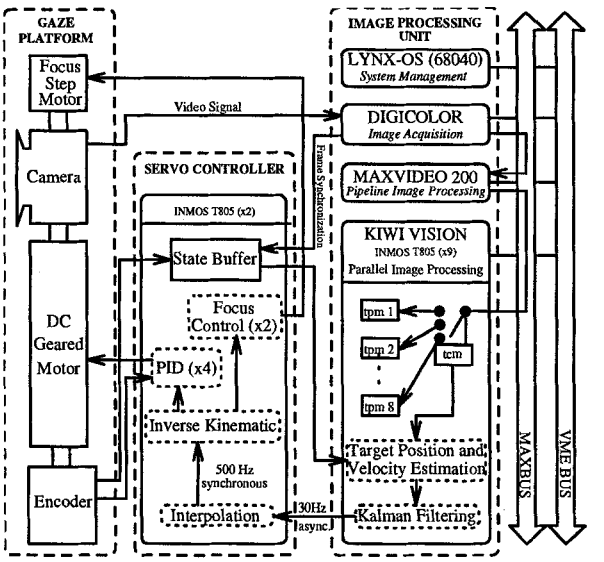


Fig. 5. Overall architecture of the tracking system

The lens: The projection curve has concrete meanings in terms of both useful vision algorithms and the correspondence with the human visual system. The projection curve of the transition region plays a double role; (1) To smoothly connect the fovea and the periphery, and (2) To achieve a meaningful image transformation. In order to cancel the shift terms in the peripheral curve and also to use the transition region as a logarithmic conformal mapping, the acquired image must be transformed from a rectangular to a polar coordinates. Real time digital warping is now an established technique and dedicated hardware (ex. MINIWARPER from DataCube) is commercially available.

Combining with a polar CCD: We propose a promising alternative design: Combining a pure spherical projection lens and a space-variant polar scan CCD. This design has the possibility to achieve the same functionality as our lens at a higher quality. It will also solve the brightness variance problem, because the wide angle lens and the space variant CCD will have opposite brightness variances and will compensate each other.

Calibration: We have developed novel calibration methods for the lens and the platform, which will be presented elsewhere. Our lens precisely traces the designed projection curve. The reference frame of the curve is the optical axis. And because of the non-linearity of the curve, the positional accuracy of an image point estimation is greatly affected by the accuracy of the real image center on the screen. This is quite different from standard lenses. Our lens calibration method measures the image center position precisely by just swinging the gaze while tracking a pair of environmental features without any special calibration

objects. We have also developed a method to precisely calibrate the mechanical joints of the platform using a similar procedure.

Vision algorithms: Currently we are developing/analyzing useful vision algorithms for our system, such as stereo tracking, motion analysis, and object recognition. This is an exciting work because of the unique property of our lens.

Acknowledgement

The authors are grateful to Professor Suematsu at Nagoya University, Japan, for kindly providing detailed data of his special lens.

Luc Berthouze developed a lens calibration method, Kenji Konaka greatly contributed to system integration, Kazuhide Sugimoto and Shin Nakamura contributed to the lens development.

References

1. P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics*, 3(4):517–535, 1993.
2. M. Imai Y. Furukawa and S. Tsuji. Stereo vision with a zoom lens. *Proc. Ann. Conf. of IEICE Japan*, D-536, 1992. (In Japanese).
3. K. J. Roberts J. C. Fiala, R. Lumia and A. J. Wavering. Triclops: A tool for studying active vision. *Int. J. of Computer Vision*, vol.12, no.2/3:231–250, 1994.
4. R. S. Wallace B. B. Bederson and E. L. Schwartz. A miniaturized active vision system. *Int. Conf. on Pattern Recognition*, pages 58–61, 1992.
5. P. H. Welch and D. C. Wood. Image tracking in real-time: a transputer emulation of some early mammalian vision processes. *Image and vision computing*, vol.11, no.4:221–228, 1993.
6. G. Sandini and V. Tagliaso. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing*, 14(3):365–372, 1980.
7. Y. Suematsu and H. Yamada. A wide angle vision sensor with fovea – design of distorsion lens and the simulated images –. In *Proc. IECON93*, volume 1, pages 1770–1773, 1993.
8. R. C. Nelson and J. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Trans. PAMI*, 11(10):1102–1106, 1989.
9. T. Morita, Y. Yasukawa, Y. Inamoto, T. Uchiyama, and S. Kawakami. Measurement in three dimensions by motion stereo and spherical mapping. In *Proc. IEEE CVPR*, pages 422–428, 1989.
10. Y. Kuniyoshi, N. Kita, K. Sugimoto, S. Nakamura, and T. Suehiro. A foveated wide angle lens for active vision. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2982–2988, 1995.
11. M. Tistarelli and G. Sandini. Dynamic aspects in active vision. *CVGIP: Image Understanding*, 56(1):108–129, 1992.

A Control Theoretical Approach to Active Vision

Henrik I Christensen, Jakob Horstmann & Torben Rasmussen

Laboratory of Image Analysis, Institute of Electronic Systems
Aalborg University, Fr. Bajers Vej 7, DK-9220 Aalborg East

1 Introduction

Active vision is here defined as deliberate control of sensory and processing parameters to enable efficient and robust achievement of specific/well defined tasks [12]. Active vision in its present form was defined by [7, 8] as a methodology for combining visual processing with physical behaviors.

Recently the area of active vision has been given substantial attention. However, most of the efforts have been directed at development of individual visual routines in the context of an agile camera head, while less emphasis has been placed on the integration of routines into fully operational systems. There are, however, a few notable exceptions. Early work on integration has been reported by Krotkow [10]. In this work disparity and accommodation cues are integrated to enable reliable estimation of depth. That work did not include any temporal processing due to the characteristics of the UPENN camera head. Later research by Clark & Ferrier [4] was inspired by psychophysical findings by Carpenter [1]. In this system three different system architectures are used in response to the set of tasks: saccadic motion, fixation or pursuit. Each of the architectures are described in terms of a control theoretical model, that describes the system behavior for each of the tasks. In general very nice work, but very few experimental results were reported for evaluation of the system. Recently there has been a significant effort at KTH [3] on integration of visual behaviors. A fully integrated system has been reported, which has a very impressive performance. For this system few theoretical results have been reported in terms of the integration strategy.

It appears as though few researchers have considered the problem of providing a sound basis for multi-cue integration in the context of fixation and pursuit, when robustness is a primary issue.

In the work reported here the fundamental hypothesis is that visual processes/routines should be organized hierarchically according to their spatio-temporal scope. I.e., image level features should be used for local/static control (fixation), while 2D tracking should be used for image level pursuit processes and finally 3D tracking should be used for scene/object level pursuit. On top of this, attention should be used for saccadic motion as scene-level attention inherently is used for change in priority between different objects/phenomena in the scene. It is further argued that the integration of the different processes should be achieved through organization of the different processes in a hierarchical/multi-

loop control system. Such an organization has the advantage that the entire system may be analyzed using traditional control theory to predict and evaluate the performance of the system.

In this paper a system constructed to verify this hypothesis is described. The paper is organized according to the different processes in the system. Initially the fixation processes are described, followed by sections on pursuit and attention. Finally implementation details are considered and experimental results are reported.

2 Fixation Processes

Fixation is by nature an image-by-image level process. The aim of the process is to control the camera system so that the optical axes intersect at the surface of the object of interest. Cues for fixation can be divided into monocular and binocular. The most common monocular cue for fixation is accommodation where the depth of the focus plane is changed to intersect with the object of interest [1, 10]. A good binocular cue for fixation is disparity (obtained from stereo matching) [1, 7]. For the integration of accommodation and disparity an obvious choice is a biologically motivated architecture [1] as shown in figure 1.

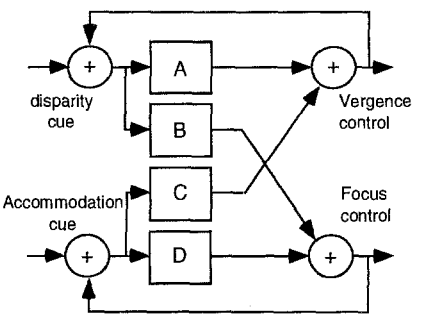


Fig. 1. Integration architecture for fixation processes

For the integration of the two cues it is necessary to have measures of "confidence" for the extracted cues. The confidence for different cues is dependent on two things: the characteristics of the algorithm for computation of the cue and the quality of the input signal (the image data). Several have studied the uncertainty for different algorithms, see [2] for examples. For the evaluation of signal quality it is characteristic that both disparity computation and accommodation require image texture to be successful. The quality of the "input signal" can be estimated from a simple spectral analysis of the central image patch. I.e, the ratio of high-frequency components compared to low-frequency components indicates the quality of the image patch for matching.

In the present implementation accommodation is computed using the Tenengrad method described by [10]. The measure computes the mean-square edge magnitude in a local neighborhood. Maximizing this measure is equivalent to obtaining a focused image. The size of the magnitude indicates at the same time the amount of image gradient (information) in the image and it can thus be used as a "signal quality" measure. The algorithmic confidence is derived from optics:

$$\Delta d_{acc} = \frac{2ACdf(d-f)}{A^2f^2 - C^2(d-f)^2}$$

where A is the aperture diameter, f is the focal length, C is the diameter of the circle of confusion, and d is the focusing distance.

For the disparity estimation, template matching is an efficient and often adequate method. Here the Sum of Squared Differences (SSD) is a good measure of matching strength. For the evaluation of signal quality the spectral characteristics is a good measure. The algorithmic confidence has been derived from simple geometric considerations:

$$\Delta d_{dis} = \frac{(d+f)^2\Delta\sigma}{fb + (d+f)\Delta\sigma}$$

here b is the size of the baseline, and $\Delta\sigma$ is the expected accuracy of the template matching.

During normal operation the disparity cue will be at least an order of magnitude better than the accommodation cue, but during occlusions the disparity cue will have no confidence and the system will be driven by the accommodation cue(s).

3 Pursuit processes

The pursuit processes are responsible for keeping the fixation point as close as possible to the object of interest during motion.

The pursuit processes can also be based on monocular or binocular cues. For the monocular case the pursuit processes for the left and right cameras, respectively, should attempt to reduce/eliminate the retinal slip.

In the binocular case an initial matching to obtain 3D information is necessary. Once 3D (depth) information is available, tracking in scene coordinates can be performed. The error measure is here distance between the fixation point and the object.

For the motion estimation a Kalman filter (or an $\alpha - \beta$ tracker) is used [11]. The matching between images may be based on structural features, such as lines, or regions. Alternatively the template matching used for the disparity estimation may be used again. To maintain a low computational complexity we have chosen to use the template matching.

The basic architecture for both 2D and 3D tracking is shown in figure 2.

For the integration with the rest of the system, it is necessary to have "confidence" measures for the tracking. Again the "confidence" measure can be divided

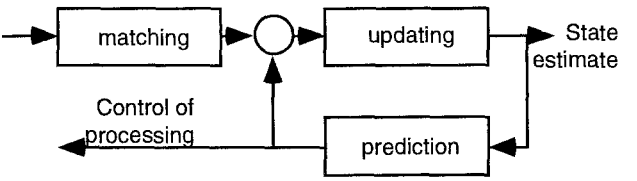


Fig. 2. Architecture for the tracking/pursuit processes

into two components: signal quality and tracking accuracy. The signal quality can be determined from the matching process, as the result/error measure from the template matching indicates the quality of the match. The match quality is dependent on the matching and the information available in the image(s). A suitable error/quality measure is thus:

$$Q_{match} = \frac{SSD_{match}}{max(SS_{image})}$$

The term SSD_{match} specifies the quality of the match between two images, and the term SS_{image} indicates the amount of edge/gradient information available in a single image. The SS measure is merely the sum of the squared edge magnitudes in a local neighborhood. The Q_{match} term is always in the range $[0;1]$. A quality/confidence measure for the tracking process is the estimate of the covariance for the prediction error. The covariance is closely related to the "validation gate" that is used in target tracking applications [11]. The variance, ${\sigma_{x,y}}^2$, along coordinate axes is used as an error measure here. To obtain a normalized error measure a ratio similar to that for the matching process is formed:

$$Q_{track} = \frac{{\sigma_{x,y}}^2}{max_error^2}$$

The two confidence measures are combined through a simple multiplication.

In terms of the Kalman filter a decision regarding the order of the filter must be taken. For most practical applications in computer vision either a first or second order filter is used. The acceleration term in a second order filter will often have problems during purely translational motion, and the results will be a noisy tracking performance. To circumvent this problem one may use a variable order filter, where the order of the filter is changed in response to the dynamics of the scene [11].

In the present implementation we are using a first order filter, for the estimation of position and velocity. The dynamics along x, y (and z) axes is considered independent, and separate trackers are implemented for each axis. The independency was enforced mainly to reduce the computational complexity, and practical experiments have demonstrated a limited gain in performance, if correlation between coordinate axes is assumed.

4 Attention/Saccades

The attention process is responsible for selection of the object to be maintained over time. This selection is typically specified by higher level processes, in response to the present interpretation of a scene. The attention mechanism may be driven by spatial and/or temporal information. In the spatial case a form of saliency is typically used. Saliency might be texture, depth discontinuities, etc. For the temporal case a frequently used cue is temporal discontinuities (initiation or termination of motion, change in motion, etc.).

For a detailed discussion of attention see [5, 6]. The latter reference has a substantial discussion of different attention mechanisms in biological and machine vision.

To demonstrate the fundamental hypothesis of the system, attention is of secondary importance. We have thus chosen to apply a very simple attention mechanism. The attention mechanism does temporal differencing to enable detection of independent motion in the scene. If such a motion is detected, the centroid for the difference region for the motion is used as the location for the next fixation (attention point).

5 Integration

The attention, pursuit and fixation processes discussed above have very different spatio-temporal scope, as indicated in the Introduction. The processes are consequently organized hierarchically according to their scope. The local (in space-time) processes form an inner loop onto which are tagged more and more global processes. In practise this implies that image-by-image processing as done in the fixation is part of the inner most loop, while image based tracking is responsible for external control of this system. Finally 3D tracking and attention is present in the outer most loop. The organization is shown in figure 3.

The different loops in the system can be coupled using summation nodes or through switching between them. In figure 3 the switching method is shown. In practice it is our experience that the switching scheme will cause discontinuities in the control, which in turn results in 'jerky' motion for the cameras. The simple summation scheme results in a more smooth motion. It is, however, important to note that the summation of different error signals is performed according to the confidence of the different measures, i.e:

$$E = \frac{Q_{2D} * E_{2D} + Q_{3D} * E_{3D}}{Q_{2d} + Q_{3D}}$$

The E-terms refer to errors while the Q-terms refer to the confidence measures. It should be noted that the 3D errors are projected into image coordinates to ensure that the summation is carried out between comparable measures.

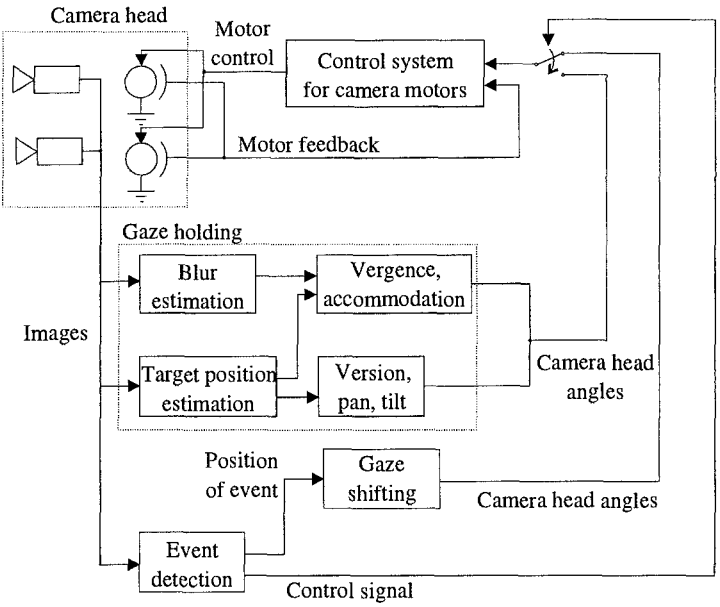


Fig. 3. Organization of the integrated system

6 Implementation

The system outlined in figure 3 has been implemented in prototype software. The test system uses the AUC camera head (figure 4), and a mix of real-time and UNIX computers.

The low level control loops (PID controllers for the motors) and the image acquisition has been implemented on a 68030 board running the real-time operating system OS/9. The low level processes are running at 25 Hz. The image processing routines are implemented on a 68040 board running UNIX V.4. Even though some of the processes are implemented as non-real time processes it is still possible to achieve a reliable processing speed of 7 Hz.

7 Experimental Evaluation

To evaluate the performance of the system shown in figure 3, a set of experiments has been designed. The full set of experiments includes fixation, 2D motion and 3D motion (with and without occlusions). These experiments are presented in detail in [9].

The performance of the system can best be demonstrated with full 3D motion, that includes occlusions. The 3D motion is necessary for the comparison of 3D tracking to 2D tracking for the general case. In addition the handling of occlusion is of interest as it represents a case where the confidence for the different control loops will be very different.

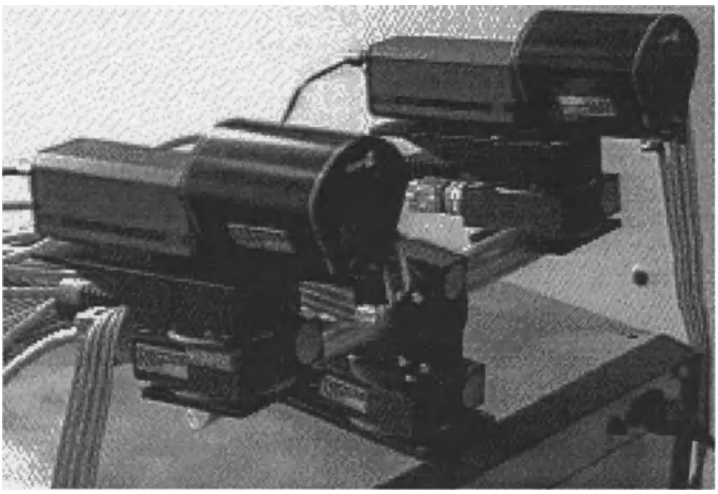


Fig. 4. The AUC camera head

For the evaluation of the system an X-Y table with an on-board target (a simple circular region) was used. This system has the advantage that it allows for generation of reproducible motion patterns (it has a repositioning accuracy of 0.2 mm over an area of 1.5×2.0 meters).

In figure 5 the Z component (depth) of the computed object location is shown for the case where tracking is based entirely on 2D cues. In the time range $[14; 16]s$ an occlusion is present. The occlusion results in a temporary loss of tracking. Once the occlusion terminates, tracking is resumed with a high accuracy. The tracking is only maintained due to the accommodation cue, which, however, is less accurate than the disparity cue.

The performance of the 3D tracking process for the same test pattern is shown in figure 6. From the figure it is easy to see that the performance is much better for the 3D tracker. This is due to the prediction in the 3D tracker. The motion pattern used in these experiments is linear in 3D, but non-linear with respect to image coordinates. The motion pattern is thus well suited for a 3D linear tracker, while a 2D tracking process is likely to encounter problems. At the same time the tracking mechanism is assisted by the inner control loops of the system.

The performance of the two systems can be compared through computation of the mean square error. The results of such a computation is shown in table 1.

For the case when no-occlusions are present the corresponding results are shown in table 2

The experiments show in general that the MSE for the 3D tracker is at least half of the the MSE for the 2D tracker. During occlusion the ratio may change to be 2-8 times better for the 3D tracking system.

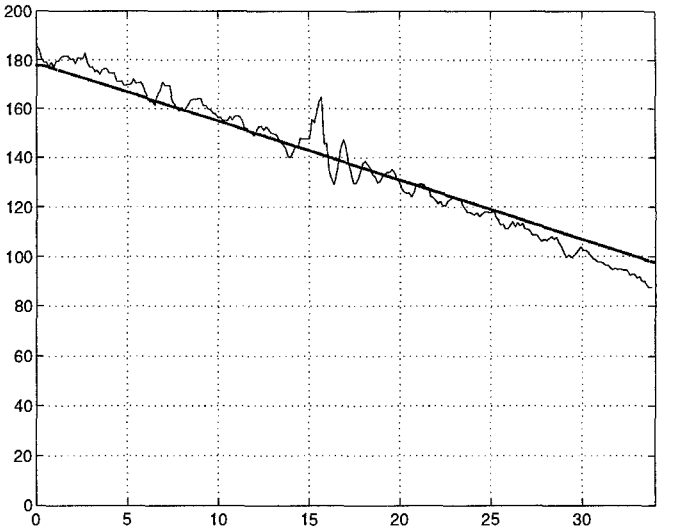


Fig. 5. Test of 2D tracking for a situation with 3D motion, where an occlusion takes place in the time frame $[14; 16]s$. Only the Z component is shown. The first axis is time [s], while the second axis shows depth [cm]. The thick line represents the actual location of the objects, while the thin line represents the measured/predicted location

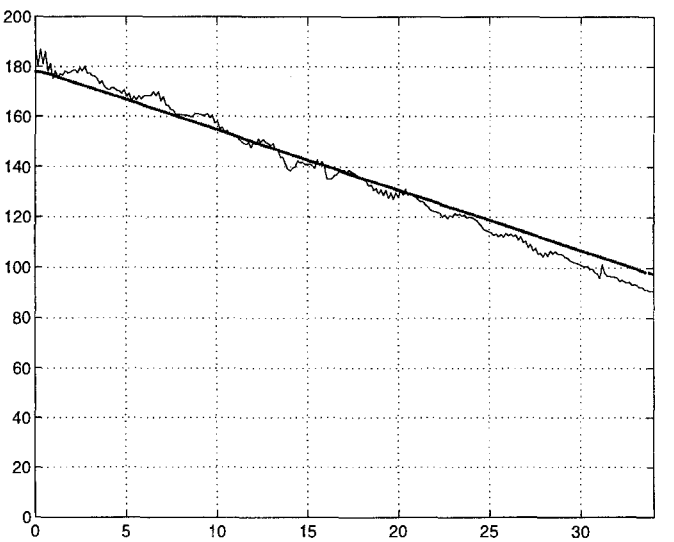


Fig. 6. Test of 3D tracking for a situation with 3D motion. Only the Z component is shown. See also caption for figure 5

Table 1. Mean square error (MSE) measures for the 2D and 3D tracking systems when an occlusion is present[cm^2]

	X	Y	Z
3-D	3.3	0.5	16.3
2-D	5.1	2.5	29.0

Table 2. Mean square error (MSE) measures for the 2D and 3D tracking systems when no occlusion is present [cm^2]

	X	Y	Z
3-D	3.1	0.6	12.6
2-D	5.5	2.1	21.2

8 Summary

Integration of active vision processes is a primary problem for construction of systems that have a robust performance. Active vision systems may be broken up into fixation, pursuit, and attention processes. These processes have very different spatio-temporal scope. If the scope is used as a guiding principle for organizing a system, a hierarchical architecture can easily be formulated. The integration strategy can in addition be based on traditional control methods, where different error/position estimates are integrated using simple summation between the control loops. In this paper we have demonstrated how image based processes, 2D and 3D tracking processes may be integrated into a coherent system. A system that is based on these principles has been designed and implemented. The system can operate at a speed of 7 Hz is fairly general settings. Experiments with the implemented system has shown that the performance of such a system is superior to the traditional 2D tracking systems.

References

1. R.H.S. Carpenter, Movement of The Eyes, London: PION Press, 1977.
2. C.S. Andersen & H.I. Christensen, Using multiple cues for controlling an agile camera head, IEEE Workshop on Visual Behaviors, pp. 97–101, Seattle, June 1994.
3. K. Pahlavan, T. Uhlin & J.-O. Eklundh, Dynamic Fixation, 4th IEEE Conf. on Computer Vision, pp. 412–419, Berlin, May 1993.
4. J. Clark & N. Ferrier, Modal control of an attentive vision system, 2nd IEEE Conf. on Computer Vision, pp. 514–523, Tarpon Springs, Fla., Dec. 1988.
5. J. Tsotsos, In the relative complexity of passive vs. active search. Intl. Jour. of Computer Vision, Vol 7, No. 2, pp. 127–141, 1992

6. R. Milanese, Detecting salient in an image: from biological evidence to computer implementation, Ph.D Thesis, Univ. of Geneva, 1993.
7. R. Bajcsy, Active Perception, IEEE Proc., Vol. 76, No. 8, pp. 996-1005, Aug. 1988.
8. Y. Aloimonos, I. Weiss & A. Bandyopadhyay, Active Vision, 1st IEEE Conf. on Computer Vision, pp. 35 – 54, London, June 1987.
9. J. Horstmann & T. Rasmussen, Active Visual Tracking using a Binocular Camera Head, M.Sc. thesis, Aalborg University, July 1995.
10. E. Krotkow, Active Computer Vision by Cooperative Focus and Stereo, Heidelberg: Springer Verlag, 1989.
11. Y. Bar-Shalom & T. Fortmann, Tracking and Data Association, Boston: Academic Press, 1988
12. M. Swain & M. Sticker, Promising Directions in Active Vision, Univ. Chicago, Tech report CS-91-27, Nov. 1991.

Active Rangefinding and Recognition with Cubicscope

Yukio Sato

Department of Electrical and Computer Engineering
Nagoya Institute of Technology
Showa, Nagoya 466, Japan
sato@elcom.nitech.ac.jp

Abstract

An active rangefinding using the Cubicscope is described in the paper. The following topics are reviewed: the principle of the Cubicscope, 3-D shape reconstruction by multiple range images obtained from actively selected viewpoints, high-speed edge detection from range image, and polyhedral object recognition by adaptive viewpoint selection.

1 Introduction

Active vision strategy must be effectively applied when there exist ambiguities for recognizing a 3-D object using a 2-D image taken from an arbitrary viewpoint. Multiple viewpoint images would reduce such ambiguities on 3-D shapes. Even if we obtain several images, however, there still remains a problem, i.e., how to reconstruct or extract 3-D shape information from such intensity images. A practical way to solve this problem is to use a rangefinder that directly grabs 2 1/2-D range images. Each pixel value of the range image indicates absolute distance from the camera to a corresponding point on the object surface. Therefore, the range image contains direct shape information about the object. Ofcourse, ambiguities about 3-D shapes can not be clarified completely, since the range image only indicates visible surface from the rangefinder. However, the active rangefinding, multiple range image grabbing by selecting the viewpoints adaptively to the object shape, will drastically reduce ambiguities in 3-D shapes.

The rangefinder we had been expecting for the active rangefinding was handy: it should be small and light enough to be installed at the tip of a robot manipulator. Furthermore, it should be able to grab range images instantaneously enough to work the robot rapidly. There had never been such a handy rangefinder until the authors developed the Cubicscope [1], which has been recently issued commercially by the CKD corporation. The size of the commercial Cubicscope is 180(W) × 85(D) × 55(H) mm and the weight is 1500g. It grabs high density range images (512 × 242 pixels) continuously within 0.3 s for every image. The robot can easily control it in the 3-D working space. Such sort of rangefinder will spread the application field where computer vision plays an important role, not only in industry but also medical, apparel, etc. In the following sections the author will review the Cubicscope briefly, and the describe some applications for active rangefinding.

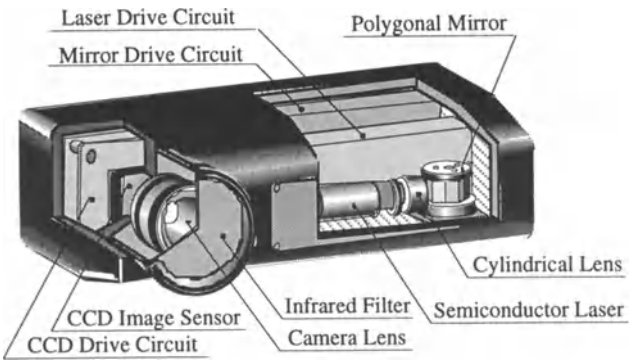


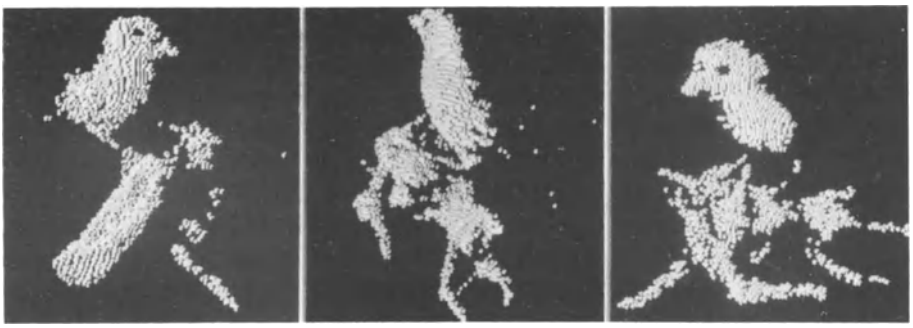
Fig. 1 Cubicscope

2 Handy Rangefinder Cubicscope

The author has proposed a method for high-speed rangefinding using a specially designed sensing device [2] and that was realized as an LSI chip by Kanade et. al. [3]. It is able to capture 1 000 range images in one second. This is one of the most potential range sensors, however using a video camera as an image input device still must be convenient because the resolution is satisfactory and it can obtain intensity images together with range images.

The prototype Cubicscope shown in Fig.1 involves a 2/3" B/W CCD camera, 30 mW semiconductor laser (680 nm), and a polygonal mirror. The method for rangefinding is based on a well-known space encoding technique. In order for generating and switching the patterns of light quickly, a semiconductor laser and a polygonal mirror are utilized. It must be one of the best illumination sources for rangefinders because it is completely compact to emit high optical power, with less heat or electrical power. Furthermore, the light can be switched at a high frequency more than M Hz. The beam of the semiconductor laser is expanded vertically to form a slit-ray, and it is scanned horizontally by a polygonal mirror. An alignment of temporal switching, which is synchronizing with the mirror rotation, yields a spatial stripe pattern of light exactly.

During the video field (1/60 s) while all the shutters of the CCD sensors are open, the slit-ray scans the scene synchronized with the video signal, and the laser switches on and off according to the temporal pattern. During the next video field (1/60 s), the signal transmits, and the another scan and image grabbing starts. Obviously, all the signals have to be controlled synchronously: the laser switching signal, the polygonal mirror, and the horizontal synchronous signal of CCD camera.



Range images obtained from some directions

Measured object

Reconstructed surface

Fig. 2 Range imge integration and 3-D shape reconstruction

This scanning and switching technique potentially allows the rapidest light pattern switching for the space encoding technique because there is no wasted time between video fields.

We use 8 light patterns to obtain 255 stripe resolutions. It requires 0.13 s for total exposure time. 16-pattern illumination is preferable to obtain stable binarization of the image, 0.27 s for the final range image. The relative measurement error for the depth is estimated as 1%.

3 Active 3-D Shape Reconstruction

A rangefinder is able to measure the spatial coordinates of points on object surfaces if the viewpoint and direction in the world coordinate system are previously calibrated. One range image partially reconstructs object surfaces which is visible to the rangefinder. Therefore, the integration of multiple range images taken from various viewpoints using the active rangefinder will reconstruct whole 3-D object shape [4]. Fig 2

indicates the process of active rangefinding and reconstruction of object shapes.

The major problems of this method are how to select the next preferable viewpoint and how to determine the end of the measurement. A 3-D object is covered by a closed surface that does not include holes or intersections. Then, it never has any edges of the surface. In our viewpoint selection strategy [5], the vision system tries to search any edges on the surface. If there exist any of them, it shifts the viewpoint to observe the gap between measured surface edges. This viewpoint selection is made adaptively to the 3-D object shape. The active rangefinder moves to the next preferable viewpoint in order to observe any unmeasured surface. The measurement will conclude if there exist no edges found on the measured object surface.

4 High-Speed Range Edge Detection

The measurement speed of the Cubicscope satisfies the requirement for the practical robot vision. Even though the range imaging sensor becomes highly practical, however, the total vision system would be evaluated unfavorable if the image processing stage is still tedious,. Therefore, the study on high-speed and practical range image processing will become one of the major subjects in computer vision research.

We now focus on the edge detection problem as one of the early range image processing. The gradient operation can be applied to detect edge candidates in a range image, as well as in ordinary intensity image processing. Since a range image contains direct information about shape, we can elaborate to extract edges directly from it. Well-known iteration endpoint search algorithm for piecewise linear curve approximation presents a hint to solve this problem; the point that gives the largest error splits the curve iteratively. This technique costs less computational time, but the approximation results are satisfactory. The extended algorithm to range image edge detection is as follows [6].

Fig. 3 shows the strategy for the iterative edge detection in a range image. Fig. (b) is polyhedral range image taken by the Cubicscope and (d) shows a boundary of the object. In Fig. (e), the furthest point from the object domain was found, and it splits the object region into triangular patches. For each plane patch, the furthest point to the object surface is searched and it splits the patch more if the error was greater than a previously determined error threshold. After the iterative calculations, several edges are extracted but they include incorrect edge lines, as shown in (f). The plane integration finally erases such lines, examining angles between adjacent patches, as in (g). The plane fitting using the least square error for each plane improves the accuracy of polyhedral approximation as in (h).

This edge extraction method is not mainly based on the local features, and the algorithm is so simple that we can expect very high-speed edge extraction. The computation time must depend on the complexities of the object shape, but in most cases, we got final results in around 0.1 s with an SGI Indy workstation. This computation time must be almost comparable with the Cubicscope rangefinding exposure time.

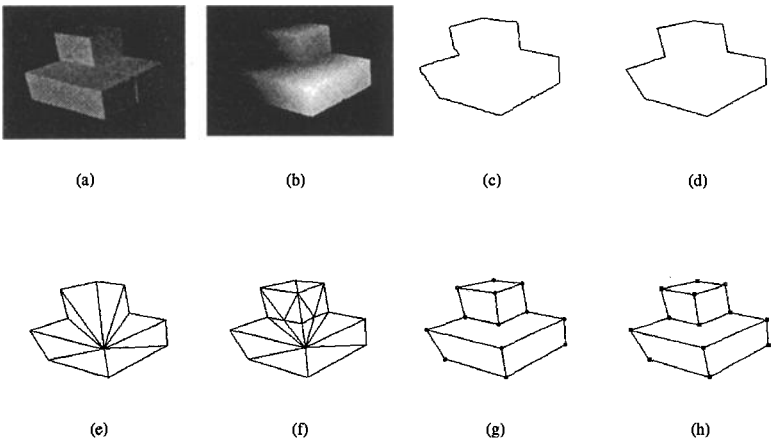


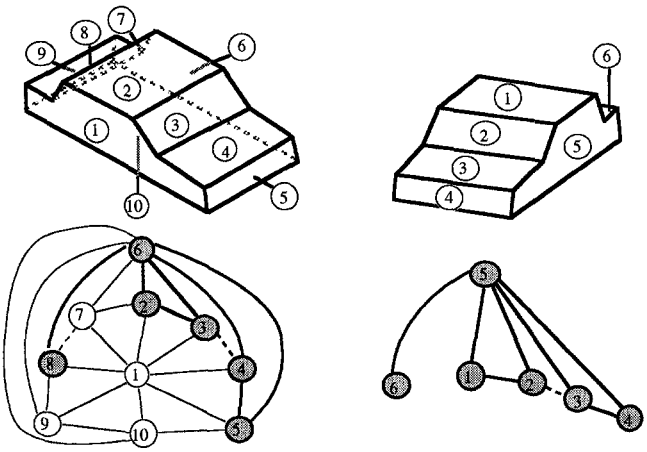
Fig. 3 Segmentation process.
(a) intensity image (b) range image (c) extracted contour (d) linear approximation (e) first division (f) polygonal approximation (g) merged regions (h)plane fitting result

5 Active View Point Selection for Recognizing Objects

The advantages ofthe active rangefinding are applicable to the object recognition. In the model-based object recognition, for example, an unknown object image is matched to models pre-stored in the computer. The problem lies also in the ambiguities about shape. Since the 2-D object image is not independent to the viewpoint, it is impossible to know how the differences were made, whether they were inherent 3-D shapes or viewpoints. However, the range image definitely matches the object surface, irrelevant ofthe viewpoint. Obviously, using range images reduces ambiguities for the 3-D object shape identification.

In our recognition strategy for the active rangefinding, a graph representation is proposed, where nodes corresponds to polyhedral planes and arcs indicate adjacent plane relations [7]. The 3-D models are previously described by model graphs (see Fig. 4). Given an unknown object, a range image is taken from some viewpoint and edges and planes are extracted from it, and the shape is represented by an object graph that examines the relations between adjacent planes. We know the object graph for the given object exactly matches to one of the model graphs. Therefore, the object recognition has been done when such a model graph was found.

If the matched model graph was uniquely located, there is no doubt the identified the object belongs to this model group. However, ifnot, the active rangefinder must shift its viewpoint and observe the object again from a different view direction. Such situations would generally happen because the proper viewpoint to observe an



(a) Model graph (b) Scene graph

Fig. 4 Graph Representation and Matching

unknown object is not a priori given in advance of the measurement. The problem here might be how to determine the next preferable viewpoint. Thoughtless viewpoint shifting is not so smart and results in wasting time. Examining the differences among the candidate model graphs, the active rangefinder will find out the viewpoint from which it can observe the most distinct shape[7].

6 Conclusion

Machine vision that interprets scenes or 3-D objects is practically important for the intelligent robots, even though we expect realistic passive vision sensors with 'shape from X'. The rangefinder has become a realistic apparatus for robot vision sensor in the sense of speed, size, and cost. We can easily induce absolute distances and 3-D shape information from range images directly. We can expect that the active rangefinder equipping the Cubicscope will dramatically solve many difficulties in computer vision research.

Acknowledgment

The author would like to thank students K. Hattori and M. Otsuki for their help in the developing the vision system.

References

[1] K. Hattori and Y. Sato, "Handy Rangefinder for Active Robot Vision," Proc. IEEE Int. Conf. Robotics and Automation, pp. 1423-1428, May 1995.

[2] K. Araki, Y. Sato, and S. Parthasarathy, "High Speed Rangefinder," SPIE, vol. 850, Optics, Illumination, and Image Sensing for Machine Vision, pp. 184-188, 1987

[3] T. Kanade, A. Gruss, and L. Carley, "A Very Fast VLSI Rangefinder," Proc. of 1991 IEEE Int. Conf. Robotics and Automation, pp.1322-1329, 1991

[4] Y. Sato and M. Otsuki, "Three-Dimensional Shape Reconstruction by Active Rangefinder," Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, New York, pp.142-147, 1993.

[5] M. Otsuki and Y. Sato, "Active 3-D Shape Reconstruction by Cubicscope," Proc. Asian Conf. Computer Vision, Osaka, pp.616-619, 1993.

[6] Y. Sato and S. Murase, "Fast Polyhedral Approximation for Range Images," Trans. Information Processing Soc. Japan, vol. 36, 10, 1995 (in Japanese).

[7] K. Hattori and Y. Sato, "Active Viewpoint Selection Algorithm for Polyhedral Object Recognition Using Range Images," Proc. Asian Conf. Computer Vision, Osaka, pp.624-627, 1993.

Real-Time Vision is NOT Frame-Rate Image Analysis

Les Kitchen ljk@cs.mu.oz.au
Tak-Keung Cheng derek@cs.mu.oz.au
Zhi-Qiang Liu zliu@cs.mu.oz.au

Computer Science Department
The University of Melbourne
Parkville, Victoria 3052
Australia

Abstract. Providing usable computer vision systems, particularly for robots, is not just matter of faster and more reliable processing of the traditional kind. It requires a quite different interaction between the vision system and its environment.

1 What's the Problem?

It is commonly felt in the computer vision community that computer vision systems are too slow and unreliable for practical real-time use. The "obvious" solution to this deficiency is to use faster hardware, and to improve both the speed and reliability of algorithms. The position we would like to put forward in this paper is that the above view is fundamentally mistaken. Certainly, improvements in speed and reliability are useful. However, such improvements alone are not likely to be sufficient to bring computer vision systems to an acceptable level of performance. Our title is intended to capture this notion, although it is something of a caricature of the situation: Processing at "frame rate" is often held up as the target or benchmark for real-time vision systems. But just doing traditional computer vision or image analysis faster and faster will not bring us usable computer vision systems.

What is more, to many workers in the field (especially those who have been doing it for a while), computer vision seems to be suffering from some kind of malaise or stagnation. We have much more powerful computers than we had 30 years ago, better understanding of the physics, better estimation techniques, more researchers, more conferences, more journals, more activity... The list goes on, and is added to every year. However, the solution of "the vision problem" seems just as far away.

We would propose that computer vision is in need of a paradigm shift, in the sense of Kuhn's *Structure of Scientific Revolutions* [13, 11]: perhaps only a minor paradigm shift, but a paradigm shift nonetheless. To some extent, this is already happening, with the advent of *active vision*, or more generally, *active perception* (sometimes even itself called a "paradigm"). We would maintain that the active perception approach (for example, [1, 2, 3, 10]), while heading in the right direction, does not go far enough, as it is usually conceived.

This paper is not intended to be the definitive critique of the state of computer vision, nor to lay out a precise program for the "revolution". Its aims are far more modest: to point out what some of the problems are with computer vision as now constructed, and make some suggestions of directions to go in. Moreover, since it is based mostly on our own research experience, the discussion will be mainly as it applies to motion tracking and object recognition.

The reader may have noticed that the last few paragraphs have not mentioned "real-time vision" explicitly; they have just spoken about computer vision, apparently in general. What of the real-time aspect? In fact, it is our opinion that computer vision is, by its nature, essentially "real-time" anyway. If computer vision is ever to provide a sense of sight for robots, then it must operate in real time, interacting with the changing (and changed) environment. Processing, no matter how sophisticated, that interprets an image without regard for the passage of time is not really computer vision. It may be reasonable to call it *image analysis*, but not computer vision. The same applies even when interpreting image sequences. We may be computing parameters, like velocity, that depend on time. But if the sequence of image frames is fixed, and no account is taken of the actual passage of time, then time is, in effect, being treated as just another spatial dimension. The analysis—now image sequence analysis—may have an additional dimension, but the nature of the processing is fundamentally unchanged.

We are not proposing here any particular definition of "real time". What we mean by it will be brought out, we hope, in the subsequent discussion. We would like to point out, though, that it is more about interactivity between the vision system and its environment than about simple processing speed. For some applications, like monitoring or surveillance, quite slow processing rates might still be compatible with real-time performance.

Traditionally, the distinction is often made between computer vision and image analysis, with the latter being used for the relatively simpler ways of extracting information from images. More sophisticated interpretation, like object recognition, is thought of as the domain of computer vision, while mere spectral pixel classification would be regarded as image analysis. (That some kinds of processing may be hard to call does not invalidate the basic distinction, since it is more a spectrum of complexity than a sharp dichotomy.)

We actually propose to re-interpret this distinction. Throughout the rest of this paper (unless qualified by some adjective like "traditional"), the term *computer vision* will be applied to those kinds of processing that could provide a sense of sight to a robot or some similar entity, like an autonomous vehicle. (Often this will be more a matter of intent than achievement, but the application still holds.) And such processing must, of necessity, have regard for the passage of time—be "real-time" in some sense. All other processing of image data (or image sequence data) can be relegated to the label *image analysis*.

Under this drawing of the distinction, a simple computer system designed say to emulate the visual abilities of an insect in flying, would be counted as doing computer vision. But traditional (static) object recognition, no matter

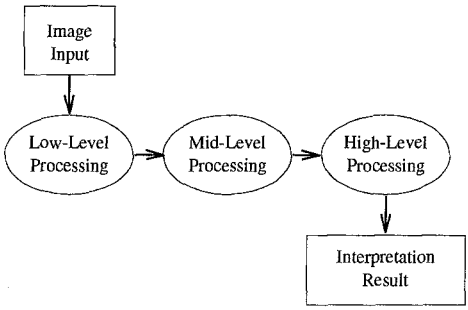


Fig. 1. "Traditional" Computer Vision.

how sophisticated, would be classed as image analysis (although admittedly of a quite complicated sort).

We do not want to act like Lewis Carroll's Humpty Dumpty and redefine terms on our whim to mean whatever we want them to mean. However, there is a valid distinction to be made, and we feel that this drawing of it better captures what computer vision is (or should be) about. Those unhappy with this particular terminology are welcome to devise their own, but this is what will be used here.

Readers of this paper may already be convinced of the importance of these considerations about real-time vision. In that case we may be preaching to the converted. However, we hope that such readers may still learn something from the issues raised and suggestions made. For there is still plenty of computer vision research done that is incompatible with real-time interactivity. We are not implying that all computer vision research should be explicitly about real-time aspects. Many issues can be best studied in isolation. However, it is of little use to develop methods that in principle could not be adapted for real-time use.

2 "Traditional" Computer Vision

What might be called "traditional" computer vision usually follows a pattern like that sketched in Figure 1. Some image input is presented to the system. (It may be a single image; it may be a stereo pair; it may be an image sequence.) The system transforms this input "upwards" through several stages of processing: Low-level stages of image preprocessing and primitive feature extraction; mid-level stages of perceptual grouping and physical parameter estimation; high-level stages of recognition and interpretation. Finally, out the end pops some "result". Of course this is a simplification: In a complex system, each of these notional levels may be broken up into intermediate levels, and there may be feedback

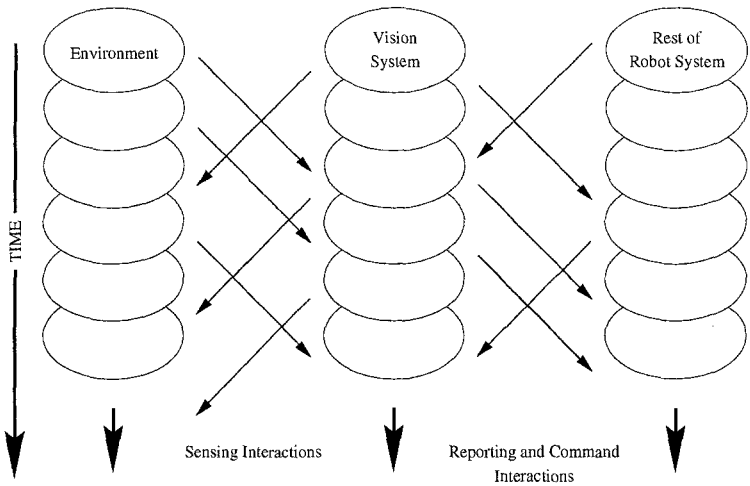


Fig. 2. Interaction in Time.

paths for data and control. However, the diagram captures the essence of such processing.

As we know, it is not unusual for many minutes of processing (or in earlier days, many hours) to be spent on the interpretation of image data that would represent only a fraction of a second of the visual sense input impinging on a real robot. This is clearly unsatisfactory. And merely speeding up processing, whether by faster hardware, parallelism, or more clever algorithms, will not really solve the problem, for the problem is a deeper one, and not fundamentally one of speed. It's as if someone on a leaking boat decided that the best strategy was just to bail out the water faster and faster. Sure, it might work, but the best solution is to fix the leak.

The real problem is that the vision processing is set up—even conceived of—as being atomic, at least as far as its input-output behavior is concerned. The input goes into the system, and there's nothing more to be said until the result comes out the end. It's reminiscent of old-fashioned batch data processing.

3 Real-Time Vision

For the vision system of a robot, the input/output behavior—or really input/output interaction—is much more complicated. It's more like the situation in Figure 2. There's the vision system itself, which is not monolithic, and may be only one of several sensory systems. There's the external environment, and

there's the remainder of the robot system (planning, control, navigation, etc.), whose detailed internal structure need not concern us here.

Between the vision system and the environment *sensing interactions* take place, in both directions. The vision system receives visual input, but has to take actions to acquire that visual data (pointing the cameras, adjusting focus and aperture, grabbing frames,...). This is basically what active vision is about. (This ignores that some of the actions, say to reposition the robot's viewpoint, may have to be mediated through the rest of the robot system.)

Between the vision system and the rest of the robot system occur interactions of reporting and control. Requests can come from the rest of the robot system for the vision system to obtain certain information, to which the vision system can respond. However, the vision system may of its own accord alert the rest of the robot system to important visual events, such as a looming collision.

All these interactions happen as everything moves forward through time. They take place over uncertain communication channels, and are subject to error and delays. Visual data acquired may only be partial. Results reported may be incomplete or tentative. Limited processing resources (and other resources) have to allocated according to the priorities of the tasks to be performed. Requests may be pre-empted or countermanded by other requests.

For example, in order for a robot to carry out a task like "Pick up the red coffee mug and place it on the table", the robot's vision system must carry out a coordinated set of sub-tasks. It must first locate the coffee mug. It must find free space for the robot's arm to move through. It must monitor that space for unexpected obstacles while the arm is moving. It must provide positioning feedback as the hand closes on the mug. It must verify that the mug has been properly grasped. And so on...

These considerations have led us to develop our own approach to vision, which we call "real-time task-directed robot vision". It is based on a conceptual architecture of a network of interacting computational agents, and makes use of a number of important principles, such as model-based prediction and verification, foveation (focus of attention), multi-resolution processing (both spatial and temporal), and asynchronous parallel (pipelined) computation. Model-based prediction and verification is the most important of the processing principles: It is an effective way of maintaining a model of the environment. We use the model to make predictions about the world, and then attempt to match those predictions. If the predictions can be matched, that serves to confirm the model. If not, then the error can be fed back to correct the model. This processing cycle is shown in Figure 3. Note the double arrows between the "compare" bubble and the "data from world" bubble. This is because the predictions drive an active process of extracting the relevant data from the environment.

Some of the earlier stages of this work have been described in [8, 7, 9]. More recent work is in [5, 4, 6, 12, 14]. So far the approach has really only been applied to problems of visual motion tracking. The approach, however, we claim is a quite general way of doing computer vision. As an example, the system diagram for the 3DFBT (*3D Feature-Based Tracker*, which does 3D motion tracking using stereo)

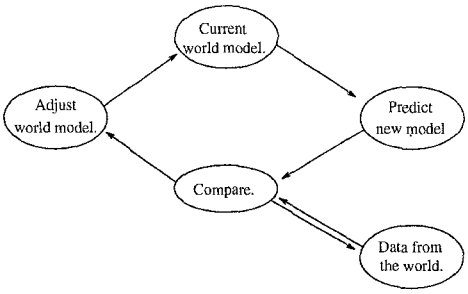


Fig. 3. Prediction/Verification Cycle.

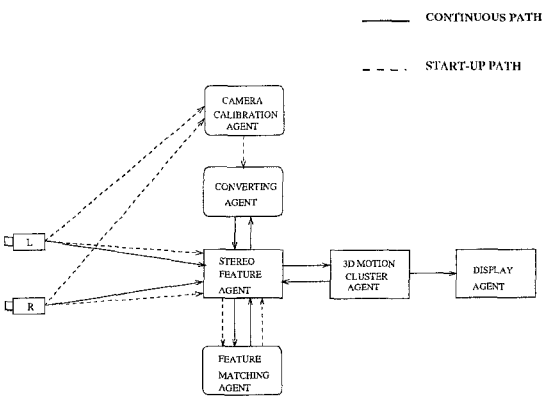


Fig. 4. 3D Feature-Based Tracker, System Diagram.

is shown in Figure 4. For the most part, each agent runs on a separate processor. The arrows between the agents show the communication paths, dashed for the start-up, initialization processing, and solid for the ongoing continuous tracking.

4 Real-Time Object Recognition

A richer domain than motion tracking is object recognition. As usually conceived, object recognition follows the traditional pattern of processing stages described

earlier: An input image is fed into the system, and, some time later, the object recognition output emerges. Typically, the output contains some identification of the seen object as belonging to some known class of objects, along with some correspondence back to the input data, and possibly object pose parameters.

We will, where necessary for clarity, make the distinction between an *object type* and an *object instance*. If there are two cups sitting on a table, there is only one object type present, "cup", but two object instances, the individual cups. (Where the distinction does not matter, or is clear from context, we will simply say "object".)

There are, of course, many variations on object recognition: The system may be able to recognize only a single kind of object; it may be able to recognize any one out of some set of "known" objects. It may be able to recognize only a single object instance in each input; it may be able to recognize multiple instances. It may only be able to recognize visually isolated objects; it may able to recognize objects amidst clutter and occlusion. But the variations all follow this overall pattern, or paradigm.

But this, we argue, is the wrong paradigm. (It is not wrong in any moral sense, it is just the wrong paradigm for achieving satisfactory performance from computer vision systems.)

Humans in general do not spend much time recognizing objects from still photographs in photo albums. It's an interesting ability, and of some use, but almost all of our daily object recognition works differently. Many creatures, like dogs and cats, have quite excellent capabilities for visual object recognition, but can make no sense of such little colored pieces of paper.

In normal situations, humans (and robots) would have extended encounters with objects in their environment. During such an encounter, object recognition can run concurrently with other processing, perhaps yielding usable partial identifications along the way. The object, or camera, might move, providing new data about the object from a different view point. The new data needs to be integrated with the ongoing recognition process. Furthermore, the robot system can actively change the camera view point, or move the object with its manipulator, or invoke some other sensor (like a tactile sensor), in order to resolve ambiguity about the object identification. These issues are partly addressed by active vision, but not completely.

What does it mean to "identify" an object? At a superficial level, it means to attach the label for some object type to the visual input coming from some object instance of that type. In reality, the situation is far more complicated than this. The space of object types really forms a (tangled) hierarchy of object classes. One might be able to recognize a certain object as a penguin, without being able to name what species it belongs to, let alone saying which individual it is. There exist parameterized families of objects, such as cylinders of varying height and diameter. An observer may have different encounters with the same individual object, in which it may not be possible, or even necessary, to recognize whether it was the same individual or just another member of the same class: *Was that the same penguin as I saw yesterday?* The simple object type/instance

distinction will suffice for many purposes, but by no means all.

In the remainder of this section, we will consider some issues in real-time object recognition.

Recognition Context A computer vision system should also be able to exploit the context of what objects are likely to appear. This is not the same as a special-purpose recognition system tailored to recognize only a few objects, because the context may change dynamically from moment to moment. For example, in a toy shop, a toy animal is much more likely to occur than a real animal, and this expectation should bias our interpretation of sense input that would otherwise be ambiguous. At one extreme, object constancy provides a very strong form of context: Once an object is recognized, it need not be recognized all over again for every input image frame, provided its location can be tracked over time (a much simpler task than recognition). The context also sets the requirements for the specificity of object identification: in many situations only a partial recognition or general classification of an object may suffice. Standing beside a road, you might be said to be "recognizing" the approaching vehicles. But what's going on will be different, depending on the circumstances: If you are trying to cross the road, it may only be necessary to classify the vehicles into general classes like car, bicycle, bus, truck, tram, in order to plan an avoidance strategy based on inferred weight, speed, and manoeuvrability. However, if you are waiting for a friend to pick you up, it is necessary to recognize only your friend's particular car. It is rarely necessary that all objects in the environment be exhaustively and specifically identified.

Dynamic Perceptual Features Most object recognition systems use a static set of perceptual features extracted from a single image. In a dynamic environment, this is inappropriate. By the time the recognition process has completed, the object (along with its features) may well have moved elsewhere, so the match back into the image would be meaningless. Indeed, because of processing constraints, it may not even be possible to extract a complete set of perceptual features from a single image frame. What is needed is some subsystem (or *agent*) that can maintain a dynamic representation of perceptual features. Within this representation, each feature has an identity, and can be accessed by the recognizer in a consistent way. On the other side, this agent would interface with the lower-level agents in order to access the incoming image data to keep the feature representation up to date.

Recognition Actions Also because of the static nature of the input, the only "actions" a traditional object recognition system can take are to make matches to the input features. However, in the case of real-time vision, the choices are much more open. For example: Some internal state of the recognizer may represent a certain ambiguity about the possible object seen, in that it could be any one of a number of objects or views. The traditional recognizer could try to resolve this ambiguity by further consideration of the perceptual features extracted from one view. However, the most effective way of resolving an ambiguity between

two object types that look similar from a certain viewpoint might be to look at the object from a different viewpoint. (A cup may look like a vase until you look at the other side to check for the presence or absence of a handle.) This requires some kind of cost/benefit analysis to decide whether it is more effective to continue matching on the current image input or to seek new input. A new viewpoint could be obtained just by switching to another camera, or by physically moving the current camera, or by turning the object with a manipulator, or (if the object is moving predictably) merely waiting until the object turns itself. And such sensing actions are not restricted to vision, but could call for the use of tactile or range sensors as well. This provides a framework in which to incorporate active vision and multisensory input into the recognition process.

Partial Recognition and Natural Object Classes In the domain of real-time vision, a recognizer may not always have the luxury of running straight to completion—it may be interrupted by other more urgent processes competing for the available processing and communication resources. The state of the recognition process needs to saved, so it could be resumed at some future time when resources became available (provided the object was still of interest).

However, looked at in another way, an intermediate state of recognition does correspond to a class of objects/views—those reachable from this state. Reaching such a state in the recognition process amounts to a partial recognition, which may still be a useful result for guiding the behavior of the rest of the system. For example, the class of objects corresponding to some intermediate state may include none of the objects the robot regards as dangerous. The recognition system at this point could notify the rest of the system that the object is (probably) not dangerous, even though it may have no final identification of the object. This may cause the system to suspend or abandon the recognition process, unless there is some other reason for settling the identification. At the moment, the intermediate states are often just accidents of the recognition strategy. The results of partial recognition would be more useful if these states could be made to correspond closely to the natural object classes of the domain.

Weak Learning Ultimately, we would like a vision system to be able to learn about objects from its environment, and incorporate what it learns into the representations it uses for recognition. That is beyond the scope of this paper. However, there is a weak form of learning. Suppose the system encounters an unknown object. It will of course fail to recognize it (or should). But, unless the object is very strange, it will have some similarities to objects the system knows about, and therefore the recognition process will proceed as far as some intermediate state. (The difference from partial recognition being that here it is not possible to proceed any further with the recognition.) Now, even if an object cannot be identified, it is still useful if the system can know when it sees that same object again. This can be achieved, in part, by suitably annotating the state where the failure occurred. If the same object is encountered again in the future (at least from a similar viewpoint), then it will presumably fail at the same state, in the same way, and this can be checked for.

5 Conclusion

In this paper we have tried to make the point that most traditional computer vision approaches are not suitable for doing real-time vision. Real-time vision require much more complicated and dynamic interaction between a vision system, its environment, and the rest of the robot of which it is a part.

Acknowledgement The work reported here is supported in part by the Australian Research Council, Large Grants Scheme.

References

1. J.(Y.) Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1:333–356, 1988.
2. R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76:996–1005, 1988.
3. D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.
4. Tak-Keung Cheng, Les Kitchen, and Zhi-Qiang Liu. Feature replenishment for long-term visual motion tracking. In *Third International Computer Science Conference: Image Analysis Applications and Computer Graphics*, Hong Kong, December 1995.
5. Tak-Keung Cheng, Les Kitchen, and Zhi-Qiang Liu. Kalman filtering and clustering for 3D motion tracking using stereo. In *Second Asian Conference on Computer Vision*, Singapore, December 1995.
6. Tak-Keung Cheng, Les Kitchen, and Zhi-Qiang Liu. Multi-agent 3D motion tracking and segmentation using binocular stereo. In *Robots for Australian Industries: 1995 National Conference of the Australian Robot Association*, pages 272–287, Melbourne, July 1995.
7. James Cooper and Leslie Kitchen. Multi-agent segmentation for real-time task-directed vision. In *Fourth Australian Joint Artificial Intelligence Conference*, pages 729–739, Perth, Western Australia, November 1990.
8. James Cooper and Leslie Kitchen. Issues, architectures and techniques in real-time vision. In *Proc. Conf. on AI, Simulation and Planning in High Autonomy Systems*, pages 218–224, Perth, Western Australia, July 1992.
9. Michael Corbett, Svetha Venkatesh, and James Cooper. An occultation-based motion segmentation technique. In *Second Singapore International Conference on Image Processing*, 7–11 September 1992.
10. E.D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision Applications*, 1:223–240, 1988.
11. Ian Hacking, editor. *Scientific Revolutions.* Oxford University Press, 1985.
12. L. Kitchen, T.-K. Cheng, and J. Cooper. A distributed systems architecture for real-time computer vision. In *Parallel and Real-Time Systems Workshop (PART'94)*, Melbourne, July 1994.
13. Thomas S. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, 2nd edition, 1970.
14. C.-Y. Tang, L. Kitchen, Y.-P. Hung, and Z. Chen. Visual tracking of 3D motion using stereo. In *Computer Vision, Graphics and Image Processing Workshop*, Taipei, Taiwan, August 1994.

Stereo Vision

Active Vision Based Stereo Vision*

Ce. WANG, Zhanyi. HU, and Song De MA

National Lab. of Pattern Recognition, P. O. Box 2728, Beijing 100080, PRC.
E-mail: huzy@prlsun5.ia.ac.cn

Abstract: In this paper, a new stereo method is introduced which, unlike to the traditional ones, does not involve image points correspondence. By controlling a pan-tilt-translation camera platform to move along an axis perpendicular to the camera optical axis, and equidistantly take multiple (M) images during the camera movement, it is shown that M image points coming from a same object point will lie on a straight line in the virtual image, which is constructed by combining the M Epipolar lines of the object point in the image order, and that the depth of this object point will be proportional to the slope value of the straight line. Thus under the active vision paradigm[1], the problem of determining the depth of an object point is converted into a much easier one of calculating the slope value of a straight line in virtual image. The experiments showed that the proposed stereo vision method is robust, and is especially adequate for qualitative vision applications.

1 Introduction

Stereo vision [2][3][4][5][6] is a widely used technique to get depth information of object in computer vision field. There are many relevant techniques in the literature which can be roughly divided into two classes, feature based stereo vision and optical flow based stereo vision. For optical flow based stereo vision, its real applicability is severely hampered by its inherent heavy computational load. In addition, optical flow determination itself is in essence an ill-posed problem and some heuristic constraints must be added to obtain a unique solution in practice. For feature based stereo vision, the problem of how to extract appropriate features and how to determine the feature correspondence has long been a challenging one, and is nothing easy to be solved for. Hence, how to find out a robust stereo vision method is of constant interests in computer vision community. In this work, we introduce an active vision based stereo vision method which converts the problem of depth determination into a much easier one of calculation of the slope value of a straight line in virtual image. The major novelty of the method is that it does not involve any feature correspondence, and consequently the robustness of the method is substantially improved as it was shown in the experiments in Section 4. The method is elaborated in details in the next section.

2 Stereo Vision Without Correspondence

2.1 Pan-Tilt-Translation Camera Platform

In order to facilitate the subsequent discussions, the pan-tilt-translation camera platform used in this work is first introduced.

---

* This work was supported by the Chinese National Foundation of Sciences and the Chinese National High Technology Program

As shown in Fig1, a CCD camera was mounted on the camera platform with the camera optical axis (Zc) parallel to the Zp axis. The camera platform can independently and simultaneously move along the Xp and Zp axes, and can also rotate around the Xp axis(tilt axis) and Yp axis (pan axis). The precision of translations is ±0.01 mm, and that of rotations ±0.01 degree. In Fig1, (Xp,Yp,Zp) denotes the platform coordinate system, and (Xc,Yc,Zc) the camera coordinate system. At the initial stage, i.e., without neither translation nor rotation, the relationship between these two coordinate systems is:

$$[Xp\ Yp\ Zp]^T=[Xc\ Yc\ Zc]^T+[0\ Dy\ Dz]^T$$

where Dy, Dz are constants.

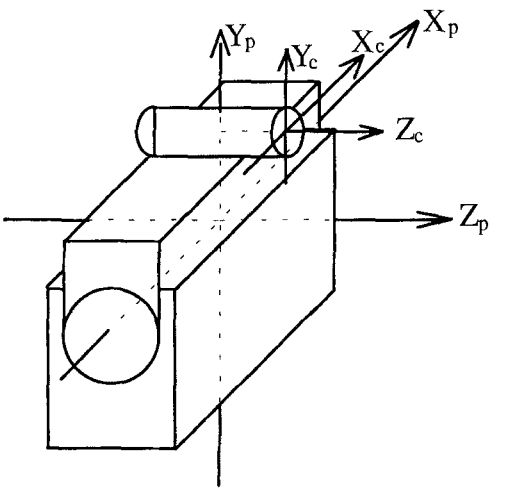


Fig1: Pan-tilt-translation camera platform

2.2 Basic Principle

Prior to any discussion, we first introduce a new term, namely a virtual image.

Let both the pan-angle β and the tilt-angle α equal zero, by controlling the camera platform to move along the Xp axis and equidistantly take multiple (M) images of size N×N during the movement, then a virtual image of size N×M is the one which is constructed by combining in the image order the M Epipolar lines of a same object point. Since the two axes of a virtual image represent two different things in nature, more specifically, one is the pixel number, and the other is the image number, hence such a combination of Epipolar lines is named as a virtual image. An example is shown in Fig2

Theorem

M image points coming from a same object point must lie on a straight line in the virtual image associated with this object point, and the depth (d) of this object point in the camera coordinate system is proportional to the slope value (k) of the straight line. i.e.,

$$d=K_0k \qquad (2.1)$$

where K_0 is a constant which is a function of the camera's intrinsic parameters, but independent of object points.

Proof

Let (X_r,Y_r,Z_r) being a object point in the camera coordinate system, and δ the displacement of the camera platform long the Xp axis between any two consecutive images (Note: hereinafter, this displacement is named as camera displacement interval), then the image point of this object point in the ith image will be (under the pin-hole camera model):

$$\begin{cases} x_i=f(X_r-i\delta)/Z_r \\ y_i=fY_r/Z_r=\text{const.} \end{cases}$$

after digitalization,

$$\begin{cases} [x_i]=fs(X_r-i\delta)/(sZ_r) \\ [y_i]=fsY_r/(sZ_r)=\text{const.} \end{cases} \tag{2.2}$$

where ($[x_i][y_i]$) is the number of pixels, and s is a resolution factor.

(2.2) indicates that the M Epipolar lines of a same object point are horizontal lines with a same constant y coordinate, as shown in Fig2.

By combining the M Epipolar lines y_i=const. i=0,1,2,..M-1 into a virtual image V(i,j), i=0,1,2,...N-1, j=0,1,2,...,M-1, we know that:

$$Z_r=K_0\frac{i-j}{[x_i]-[x_j]} \tag{2.3}$$

(2.3) indicates that all the M image points of a same object point lie on a straight line in the virtual image, and that the depth of this object point is proportional to the slope value of the straight line. An example with real images is shown in Fig4.3.

Now we have all the necessary materials to present our active vision based stereo vision method, which essentially consists of the following four steps:

<1>: Equidistantly take multiple images by controlling the camera platform to move along the Xp axis;

<2>: Build virtual images;

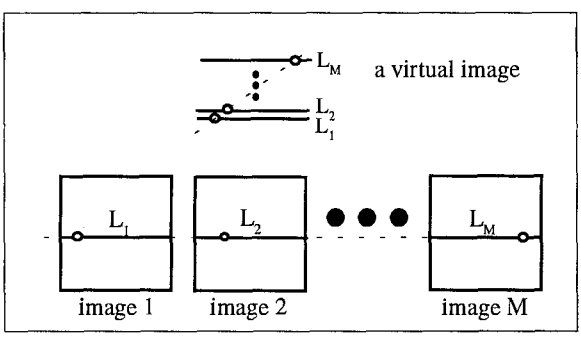


Fig2: A virtual image is constructed by combining in the image order the M Epipolar lines of a same object point

<3>: Extract straight lines in virtual images;
<4>: Based on (2.1), calculate the depths of object points.

From (2.3), we know the proposed method possesses the following two advantages over the traditional ones:

<1>: The method converts the problem of depth determination into the one of calculating the slope value of a straight line, the later is much easier to be done.
<2>: The method does not involve any image points correspondence, but it does need to know the Epipolar lines of a same object point. From (2.2), we know that if the camera platform moves along the Xp axis, all the Epipolar lines of a same object point are horizontal ones with a same constant y coordinate, thus the determination of such Epipolar lines does not pose an additional problem in this case.

2.3 Principle Extension

In the preceding section, the camera movement is restricted to a pure translation along the Xp axis and both the pan angle β and the tilt angle α are required being zero. In this section, it is shown that the basic principle introduced in the preceding section still holds even if one or some of these conditions are not satisfied.

As shown in [7], the fundamental requirement of the method is that during the camera movement, the Z_r coordinate of an object point (X_r, Y_r, Z_r) under different camera coordinate systems must keep unchanged. Thus we have:

<1>: During the camera movement, the pan angle β must be set to zero. Otherwise, the fundamental condition of the method will be violated. This problem will be discussed in details in the next section.
<2>: During the camera movement, the tilt angle α can be a fixed value. However, it should be emphasized that although the tilt angle α can be an arbitrary value, it must keep unchanged during the camera movement.
<3>: With the pan angle β=0 and tilt angle α being a fixed value, the proposed method still works while the camera moves along a straight line in the Xp-Yp plane rather than along the Xp axis. In this case, the Epipolar lines of a same object point are no longer horizontal lines, but straight lines with a same constant slope value. Readers are referred to [7] for details. In addition, it should be noted that if the camera moves along a curve rather than a straight line in the Xp-Yp plane, the method will no longer work, the reason is that in this case, it will be difficult to determine Epipolar lines of a same object point in different images.
<4>: Finally, it is worth mentioning that amongst the six intrinsic camera parameters, only those concerning radial lens distortion will affect the basic principle in Section 2.2.

3 Error Analysis

In this section, the following two problems will be discussed:

<1>: Errors due to non-zero pan angle;
<2>: Minimum depth resolution.

3.1 Errors Due to Non-Zero Pan Angle

As we have indicated in section 2.3, the proposed method requires the pan angle setting to zero. However, in practice, it is difficult to do so. If the pan angle is not zero but with

a small value β, the Epipolar lines of a same object point will no longer be horizontal ones, and consequently the proposed method becomes invalid theoretically. But if β is quite small, the proposed method can still be used with some minor modifications.

In what follows, it is shown how the pan angle β will affect the basic principle presented in section 2.2.

Assuming the camera moves along the Xp axis and M images are equidistantly taken with the camera displacement interval being δ. Denote the camera coordinate system at i^{th} image by C_{i0} when $\beta=0$, and by $C_{i\beta}$ when $\beta\neq 0$, then:

$$C_{i\beta}=\begin{bmatrix} X_{i\beta} \\ Y_{i\beta} \\ Z_{i\beta} \\ 1 \end{bmatrix}=\begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} C_{i0}$$

$$=\begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & -i\delta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix} \tag{3.1}$$

$$=\begin{bmatrix} X_r\cos\beta + Z_r\sin\beta - i\delta\cos\beta \\ Y_r \\ -X_r\sin\beta + Z_r\cos\beta + i\delta\sin\beta \\ 1 \end{bmatrix}$$

where$[X_r\ Y_r\ Z_r]^T$ is the coordinates of an object point in C_{00}, then the image coordinates in the ith image will be:

$$\begin{bmatrix} x_{i\beta} \\ y_{i\beta} \end{bmatrix} = \begin{bmatrix} fX_{i\beta}/Z_{i\beta} \\ fY_{i\beta}/Z_{i\beta} \end{bmatrix}$$

$$=\begin{bmatrix} f\dfrac{X_r\cos\beta + Z_r\sin\beta - i\delta\cos\beta}{-X_r\sin\beta + Z_r\cos\beta + i\delta\sin\beta} \\ f\dfrac{Y_r}{-X_r\sin\beta + Z_r\cos\beta + i\delta\sin\beta} \end{bmatrix} \tag{3.2}$$

From (3.2), it is clear that in this case $y_{i\beta}\neq$ Const.

If $\beta=0$,

$y_{i0}= f\,Y_r/Z_r$

then

$$\Delta Y_i= \left| y_{i0}-y_{i\beta} \right|$$

$$=\left| y_{i0} \right| \left| \frac{-X_r\sin\beta + Z_r(\cos\beta-1) + i\delta\sin\beta}{-X_r\sin\beta + Z_r\cos\beta + i\delta\sin\beta} \right|$$

if β is small, $\cos\beta\approx 1$, $\sin\beta\approx\beta$, then:

$$\Delta Y_i = |y_{i0}| \left| \frac{(i\delta - X_r)\beta}{Z_r - \beta(i\delta - X_r)} \right|$$

if $\Delta Y_i < \varepsilon$, we have:

$$\begin{cases} \beta < \dfrac{\varepsilon f}{(|y_{i0}| + \varepsilon)\,|x_{i0}|} & \text{if } f/(\beta x_{i0}) > 1 \\ \beta < \dfrac{\varepsilon f}{\big|\,|y_{i0}| - \varepsilon\,|\,x_{i0}|\,\big|} & \text{otherwise} \end{cases} \tag{3.3}$$

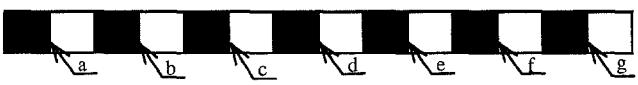


Fig3: Two experiments were designed to determine the depths of points a,b,c,d,e,f,g

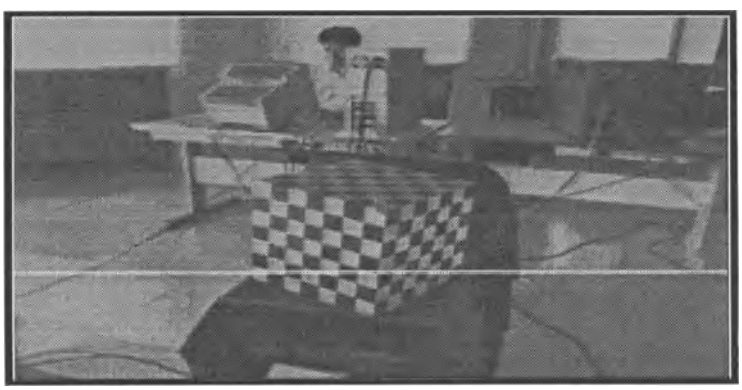


Fig4.1: First Image

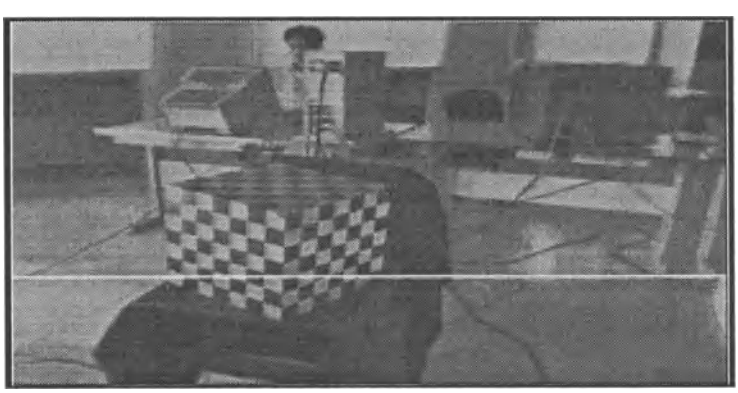


Fig4.2: Last Image

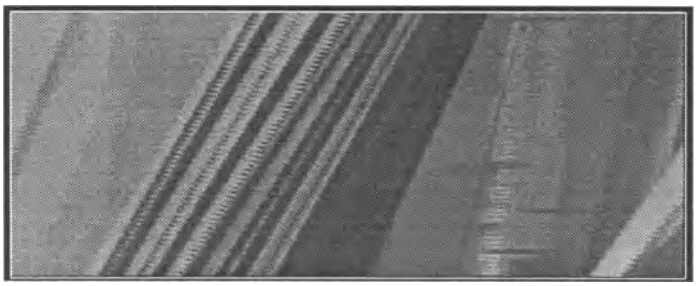


Fig4.3: A Virtual Image

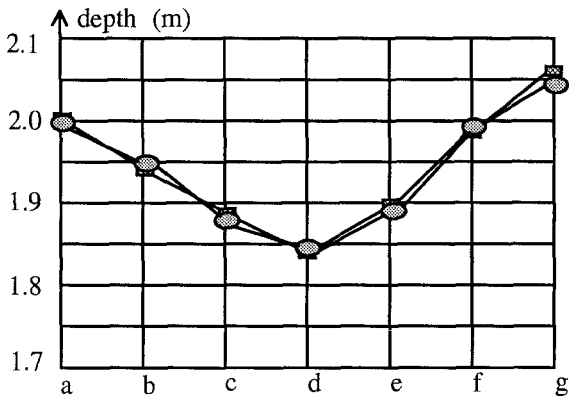


Fig5.1

Reference depth

Depth determined by the method pan angle is zero

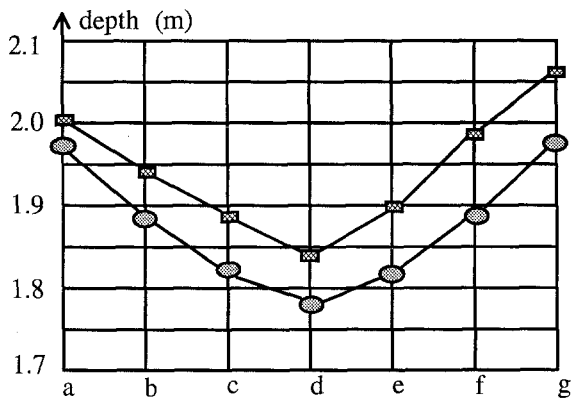


Fig5.2

Reference depth

Depth determined by the method pan angle is 5 degrees

similarly, we can obtain:

$$\Delta x_i = | y_{i0} - y_{i\beta} |$$

$$\approx f \frac{Z_r^2 + (X_r - i\delta)^2}{Z_r^2} \beta + O(\beta^2)$$

(3.3) indicates that if β is small, the basic principle introduced in section 2 can still be used but with the modification that a virtual image is composed of image bands centered on Epipolar lines with 2ε width rather than directly Epipolar lines themselves. Fig4.3 is an example where the virtual image is composed of image bands of 3 pixel width.

In summary, if the pan angle β is not zero, strictly speaking, the proposed method will become invalid. However, if β is quite small, the method can still be used except that the virtual image is composed of narrow image bands centered on Epipolar lines rather than Epipolar line themselves. The width of the image bands is application dependent, as it is shown in (3.2), it is a function of β and Z_r .

3.2 Minimum Depth Resolution

As shown in (2.1), the depth of an object point is proportional to the slope value of a straight line in virtual image. Since a virtual image contains a finite number of digital lines, the depths determined by the method are consequently discrete ones. Now the question is that among all the slope values $M_0 = \{m_i, i=1,2,....L\}$, which could be determined from an image represented by a $\{0,1,...N\}\times\{0,1,...M\}$ bit array, and for a given slope value m_0, how to determine the m_0's closest neighbor, i.e.,

$$\mathrm{Min}\{ \ |m_i - m_0| \ | \ m_i \neq m_0, \ m_i \in M_0 \ \} \qquad (3.4)$$

Such a problem closely depends on the definition of a digital line. Different definitions exist in the literature. Even for the simplest one that a straight line is determined by two different image points, as shown in [7], it is proved to be difficult to find out an analytical expression for (3.4). However, as shown in [8], a global minimum does really exist. i.e.,

$$\mathrm{Min}\{ \ |m_i - m_j| \ | \ m_i \neq m_j, \ m_i, m_j \in M_0 \ \} = \frac{1}{N(N-1)}$$

4 Experiments

The two experiments were designed to test the validity of the method. The experiments are detailed as follows:

As shown in Fig4.1, in front of the camera, a 28×28×28 cm^3 camera calibration pattern was placed. The pattern was covered with interleaved 4×4 cm^2 white and black squares. The distance between the pattern and the camera was approximately 2 m. The experiments were designed to determine the depths of points a,b,c,d,e,f,g, as shown in Fig3. Fig3 was an enlarged one for clarity. It is in fact a band of 3 lines width, and is the one included between the two horizontal lines as indicated in Fig4.1. The camera was controlled to move along the Xp axis and the camera displacement interval was 5 cm. 59 images in total were taken for each experiment. For the first experiment, the pan angle

was zero. For the second experiment, the pan angle was set to 5 degree. For both these two experiments, virtual images were composed of image bands of 3 lines width centered on Epipolar lines rather than Epipolar lines themselves. The final results were shown in Fig5.1 and Fig5.2 for the first and second experiments respectively.

From Fig5.1 and Fig5.2, it is shown that the method is sufficiently accurate and reliable. In addition, it is found that the method is especially adequate for qualitative vision. For example, for vision guided robot navigation, we only need to check whether or not exist obstacles in the front region near to the robot, hence only part of a virtual image rather than the whole image should be checked to determine whether or not exist sufficiently long lines within it. We have implemented this method in our robot navigation system, and the preliminary results are satisfactory.

5 Conclusions

An active vision based stereo vision method was introduced, which unlike to the traditional stereo ones, does not involve any image points correspondence, and converts the problem of depth determination into a much easier one of calculation of the slope value of a straight line. The experiments show that the method is accurate and robust. In addition, it is found that the method is especially adequate for qualitative vision applications.

Acknowledgment

Thanks are due to Mr. Wei WANG and Mr. Qi Fa KE for their helps during the preparation of this manuscript.

References

[1]: J.Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision", Int. J. Computer Vision, 1(4), pp.333-356, 1988.

[2]: S.T.Barnard and M. A. Fischler, "Computational stereo", ACM Computing Surveys 14, pp.553-572, 1982.

[3]: U. R. Dhond and J.K.Aggarwal, "Structure from motion---A review", IEEE Trans. on Systems, Man, and Cybernetics 19, pp.1989-1510, 1989.

[4]: X.Y.Jiang and H. Bunke, "Line segment based axial motion stereo", Pattern Recognition, Vol.28, pp.553-562, 1995.

[5]: D. W. Murray, " Recovering range using virtual multicamera stereo", Computer Vision and Image Understanding, Vol.61, No.2, pp.285-291, 1995.

[6]: J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi, "Divergent Stereo in Autonomous navigation: From bees to robots", Inter. J. Computer Vision 14, pp.159-177, 1995.

[7]: C. Wang, "Active Vision Based Stereo Vision Analysis", M.S thesis, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 1995.

[8]: T. Risse, "Hough transform for line recognition: complexity of evidence accumulation and cluster detection", CVGIP 46, 327-345, 1989.

Motion of a Stereo Rig: Strong, Weak and Self Calibration

Zhengyou Zhang

INRIA, 2004 route des Lucioles, BP 93, F-06902 Sophia-Antipolis Cedex, France

Abstract. This paper addresses different issues of motion analysis with stereovision. The stereo system can be either strongly calibrated in the classical sense, or weakly calibrated in the sense that only the epipolar geometry is known, or even not calibrated at all so we must seek for information only from the surrounding environment by moving the cameras in it (self-calibration).

1 Introduction

It is well recognized that stereoscopic cues play an important role in understanding the 3-D environment surrounding us and provide a robust way for 3-D reconstruction. In order for the 3-D reconstruction to be possible, one needs to know the relationship between the 3-D world coordinates and their corresponding 2-D image coordinates for each camera. This is the purpose of classical camera calibration, which we would like to call *strong calibration.* A wealth of work on strong calibration has been carried out by researchers either in Photogrammetry or in Computer Vision and Robotics. The usual method is to compute cameras' parameters from one or more images of an object of *known size and shape*, for example, a flat plate with a regular pattern marked on it. The object should be *visible to all cameras.* In this framework, 3-D *metric* information is available, and this is a world we are used to deal with.

Nevertheless, this technique is rather difficult to bring into operation, and the results revealed are not very stable. Many studies show that we do not always need metric information in many applications such as navigation and object recognition. This leads to use of *weakly calibrated stereo.* Given only a number of point matches between two images, we are able to estimate the *epipolar geometry* between the two cameras. This allows us to reconstruct the environment with respect to a projective basis. Although the reconstruction is not metric, a rich information is still available (e.g. cross ratio, collinearity). By introducing some simple knowledge such as parallelism, even an affine reconstruction is possible, in which the ratio of distances is conserved.

One problem with these techniques is that it is impossible to calibrate cameras on line, while they are involved in a visual task. This has motivated a number of researchers to develop online camera calibration techniques, known as *self-calibration.* The idea is to calibrate cameras by just moving them in the surrounding environment. The *motion rigidity* provides several constraints on the cameras' intrinsic parameters. They are more commonly known as the

epipolar constraint, and can be expressed as a 3×3 so-called *fundamental matrix*. From the acquired images of the environment alone, we are not only able to estimate the epipolar geometry of the stereo, but also able to achieve a metric reconstruction of the environment.

2 Strongly Calibrated Stereo

This is the most common configuration, where each camera of the stereo rig has been calibrated using techniques such as those described in [24, 7, 27] (i.e. the intrinsic parameters are known) and the relative geometry between cameras (i.e. the extrinsic parameters) is also known. Thus, at each time instant, we can perform a 3-D reconstruction (see e.g. [1]), which provides us with a local representation of the environment. The problem of motion analysis with a stereo rig can then be considered as the matching of geometric primitives in 3-D space [36]. We will also consider the motion-stereo cooperation approach which avoids 3-D reconstruction at each time instant.

2.1 Matching in 3-D Space

Here, we distinguish two situations: *short sequences* where each stereo image is a snapshot of the scene and can be quite different from other images in the sequence; *long sequences* where successive images are separated by a short time interval.

Short sequences We consider the case of two views. Given two sets of 3-D primitives observed in two views, the task of matching is to establish a correspondence between them. By a correspondence, we mean that the two paired primitives are the different observations (instances) of a single primitive undergoing motion. The matching problem has been recognized as a very difficult problem. It has an exponential complexity in general. For example, if we structure the matching as a search in the interpretation tree [9], one should examine the consistency of $(n+1)^m$ interpretations, where m and n are the numbers of primitives in the two views. Recognizing this, we need some constraints to heuristically guide the search without exploring most of the nodes of the tree while making sure a good interpretation is found at the end.

Hypothesize-and-verify method. The rigidity assumption about the environment and objects is used in most matching algorithms. We have developed an algorithm using the rigidity constraints to guide a hypothesize-and-verify method [36, 33]. Figure 1 illustrates diagrammatically the principle of our hypothesize-and-verify method. Our idea is simple. We use the rigidity constraints to generate some hypothetical primitive correspondences between two successive frames. There are a variety of choices of different types of geometric primitives. We consider here the case of line segments. Because two 3-D segments are the minimum number in order to compute uniquely the 3-D motion, the rigidity constraints

are formulated using two 3-D line segments. In [36], we have formulated a set of invariants under the rigid transformation and have shown that they are complete for 3-D line segments, in the sense that they are necessary and sufficient to guarantee the congruency of two sets of line segments. Due to inherent noise in measurements, the equalities in the rigidity constraints are not true anymore. Unlike other methods using some empirical fixed thresholds, we determine dynamically the threshold from the error measurements of 3-D data given by a stereo system. Two pairings of 3-D line segments form a matching hypothesis if they satisfy the rigidity constraints. Since we apply the rigidity constraints to any pair of segments, if we explore all possible pairs, the complexity is $O(m^2n^2)$, where m is the number of segments in the first frame and n that in the second frame.

For each hypothesis, we compute an initial estimate of a displacement using the iterative extended Kalman filter. The uncertainty of 3-D data is again taken into account in the filter. We then evaluate the validity of these hypothetical displacements. We apply this estimate to the first frame and compare the transformed frame with the second frame. If a transformed segment of the first frame is similar enough to a segment in the second frame, this pairing is considered as being matched and the extended Kalman filter is again used to update the displacement estimation. After all segments have been processed, we obtain, for each hypothesis, the optimal estimate of motion, the estimated error given by the filter and the number of matches. As one can observe, the complexity of verifying each hypothesis is $O(mn)$. Once all hypotheses are evaluated, the hypothesis which gives the minimal estimated error and the largest number of matches is considered as the best one, and its corresponding optimal estimate is retained as the displacement between the two frames. Due to the rigidity constraints, the number of hypothetical displacements is usually very small, and computational efficiency is achieved. We exploit the rigidity constraint locally in the hypothesis generation phase and globally in the hypothesis verification phase.

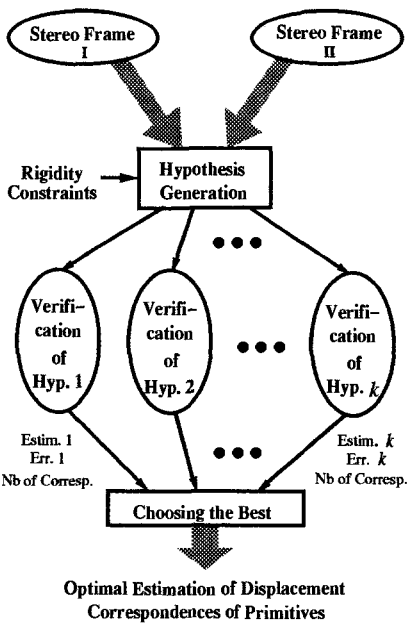
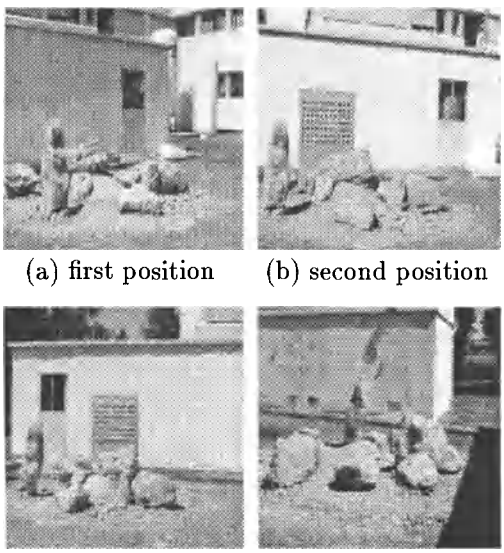


Fig. 1. Diagram of the displacement analysis algorithm based on the hypothesize-and-verify paradigm

A number of methods have been used in our implementation of the above method to reduce the complexities of both the hypothesis generation and verification stages. The interested reader is referred to [36, 33, 35] for more details. The reader is also referred to [20, 2, 3, 14] for other related works.

Iterative point matching. The information used in the above primitive-based approach is concise, and is in general preferable. However, in a natural environment such as outdoors, with the state-of-the-art techniques, it is difficult to detect robustly and localize precisely primitives [26, 17]. On the other hand, an active sensor such as a laser rangefinder or a correlation-based stereovision system [8] can be used to build a dense 3-D map, which can be considered as the samples of a surface. The surface-based approach tries to find the 3-D motion by minimizing a criterion relating the distance between two surfaces. It uses all available information. The large redundancy allows for a precise computation of the transformation between the two maps, but this approach usually requires some a priori knowledge of the transformation.

In many applications, for example, visual navigation, the motion between curves in successive frames is in general either small (because the maximum velocity of an object is limited and the sample frequency is high) or known within a reasonable precision (because a mobile vehicle is usually equipped with several instruments such as odometric and inertial systems which can provide such information). In the latter case, we can first apply the rough estimate of the motion to the first frame to produce an intermediate frame; then the motion between the intermediate frame and the second frame can be considered to be small. A surface-based method is attractive for such applications.



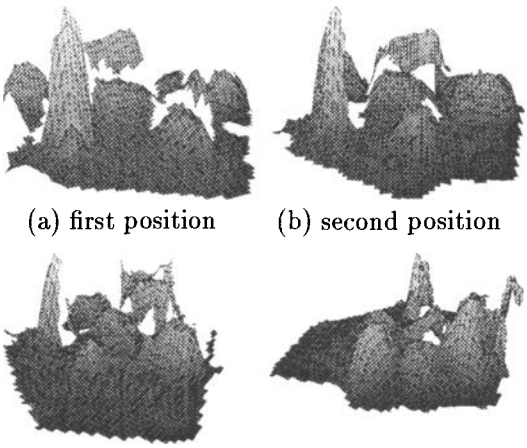
(a) first position (b) second position

(c) third position (d) fourth position

Fig. 2. Images taken by the first camera of a stereo rig from four different positions

The key idea underlying our technique described in [29] is the following. Given that the motion between two successive frames is small, a point in the first frame is close to the corresponding point in the second frame. By matching points in the first frame to their closest points in the second, we can find a motion that brings the two sets of points closer. Iteratively applying this procedure, the algorithm yields a better and better motion estimate. A statistical method based on the distance distribution is used to deal with outliers, occlusion, appearance and disappearance, which allows us to do subset-subset matching. A least-squares technique is used to estimate 3-D motion from the point correspondences, which reduces the average distance between points in the two sets. Both synthetic and real data have been used to test the algorithm, and the results show that it is efficient and robust, and yields an accurate motion estimate.

Multiple viewpoint integration (fusion). A single stereo view can only provide partial imprecise information about the environment, which is not sufficient for interpretation, recognition, or navigation. The objective of data fusion is to build a consistent, accurate, and complete description of objects or environments by combining the observations taken by the stereo system from multiple views. This is now possible because an estimate of the motion between two successive views is available with the techniques described above, which allows us to bring observed data together and express them in a commun coordinate system. An example is shown in Fig. 2 and Fig. 3 for the incremental construction of the digital elevation map (DEM) of a rock scene. We can observe how the holes in the initial DEM (Fig. 3a) are filled progressively when more data is available. The last DEM (Fig. 3d) is almost complete and the obstacles are well modeled. The reader is referred to [32, 11] for more details.



(a) first position (b) second position

(c) third position (d) fourth position

Fig. 3. Incremental construction of the digital elevation map (DEM) of the rock scene

Long sequences Although a number of constraints or heuristics, especially the rigidity assumption, have been incorporated, the correspondence problem in short sequences is still found to be very difficult. However, the problem would become much easier if long sequences of images taken at short time intervals are used. Indeed, as the time interval is small and object velocity is constrained by physical laws, the interframe displacements of objects are bounded, i.e., the correspondence of a token in the next instant must be in its neighborhood. Furthermore, objects usually move smoothly [25, 13, 23], thus the motion coherence can be used to predict the occurrence of tokens in the future, which considerably reduces the search space. The statistical data association techniques for target tracking, originally developed for radar imagery, fit well in this framework, and are already recognized as an efficient approach to solving correspondence and motion problems [4, 30].

Here, we briefly outline the technique we have developed [34]. To clarify the presentation, we call the geometric primitives being tracked the *tokens* and the currently observed 3-D primitives the *scene tokens.* The motion analysis of long sequences is carried out in a framework which consists of two levels (see Figure 4). In the figure, single arrows represent data flow, and the double arrow represents the flow of control.

The low level is called the **tracking team.** A token being tracked can be

considered as one of the team members. The token tracking process uses a model of the 3-D kinematics, instead of a model of the *position* evolution. A token is then characterized by its *position* and *orientation* in the current frame, its *kinematic parameters*, and a positive number called *support*. The support indicates the degree of support for the existence of the token, and is described in [34].

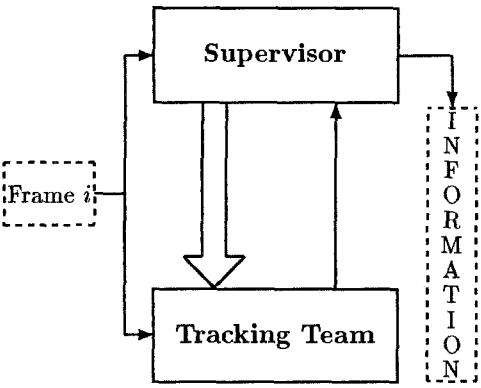


Fig. 4. An architecture for motion analysis in a long sequence

When a new frame is acquired, each token being tracked "searches" the whole frame for a correspondence. The search space can be considerably reduced by using a kinematic model and information from previous frames: before the new frame is available, one can predict the occurrence of each token in the new frame based on the kinematic model. When the new frame is obtained, one only needs to look for scene tokens in the neighborhood of the predicted position. When a match is found, the kinematics, position and support parameters are updated, The prediction and update are done by using an extended Kalman filter. The matching process is based on the Mahalanobis distance. Due to occlusion or absence of some scene tokens in the current frame, a token may not find any match in the neighborhood of its predicted position in the current frame. Of course, this phenomenon may also occur due to the disappearance of the token. To handle the occlusion and absence of scene tokens, it is necessary to hypothesize the existence of the token and continue to change its kinematic parameters according to the kinematic model and update its support. A *beam-search* strategy has been proposed to handle multiple matches based on the idea of least-commitment. The tracking of a token is thus carried out in a *prediction-matching-update* loop, and this process is performed independently for each token to be tracked, which allows a completely parallel implementation.

The high level is called the **supervisor**. It has three main functions:

- Grouping tokens with similar kinematic parameters as a single object. If there exist multiple moving objects, they can be segmented on the grounds that they undergo different motions.
- Monitoring the tracking team by detecting the following events: appearance of new tokens; false matches; disappearances of tokens; occlusion; multiple matches; abrupt changes in the motion of a token due, e.g., to collision.
- Providing information to other components of the global system. For example, in an active tracking application, one may need to control the motion of the robot or adjust the camera parameters to adapt the changing situation based on the estimation of the kinematics of the robot (ego-motion) and the kinematics and relative positions of the moving objects.

An important feature of our approach is that we build a model of the 3-D motion kinematics of each token, assuming that *it is attached to a moving rigid object.* This is different from other work on tracking where *a polynomial model of the token position* is used. Therefore, we can detect multiple motions by grouping tokens that have similar kinematic parameters, thus obtaining a segmentation of the scene into "objects" (i.e. sets of tokens) moving rigidly. See [36, chap.14] for details on motion-kinematics modeling.

2.2 Motion-Stereo Cooperation

The techniques described in the previous subsection work in 3-D space, and good results have been obtained. However, they suffer from the following two problems: (*i*) We have to reconstruct a 3-D visual map at each time instant (Considering the high redundancy of information in a temporal sequence, this is not a very efficient way for analysing a long sequence); (*ii*) The primitives only visible from one camera are not reconstructed by stereo (There is thus a loss of information). To overcome these, we have developed an algorithm which makes stereo matching and temporal tracking cooperate with each other [19, 18]. It is an iterative procedure and applies to two pairs of successive stereo images. At time t_1, a set of 3-D primitives are available (initially they are reconstructed from the first pair of stereo images), and the predicted motion is applied to each primitive. Each transformed primitive is then reprojected onto each camera (*prediction*). For each predicted primitive, the best match is searched among all observed primitives at time t_2 (*matching*). The motion estimate and the 3-D primitives are then *updated.* Finally, for those primitives which are not matched by stereo, they are reconstructed based on *motion and structure from motion*, and are added in the 3-D data base.

3 Weakly Calibrated Stereo

Strongly calibrated stereo gives much better results than a monocular vision system. However, the relative geometry together with intrinsic parameters of the cameras in a stereo system must be first calibrated, which requires the calibration apparatus to be visible from *each* camera. This procedure is not trivial to put into operation, because if the calibration apparatus is too far from the cameras, the calibration quality is poor, and if the calibration apparatus is too close, it is visible from all cameras. This section describes two approaches to this problem.

3.1 Stereo with Unknown Extrinsic Parameters

The problem addressed here is how to determine 3-D motion and structure from 2-D feature points observed by a binocular stereo rig at two different times. The relative location and orientation between the two cameras, i.e. the extrinsic parameters of the stereo rig, are not known, but remain constant during the period in consideration. The motion of the stereo rig is also unknown. We have

shown that, given projections of four or more points, we can compute analytically the relative location and orientation of the cameras, the motion of the stereo rig, and the 3-D structures of the points, up to a scale factor. Nonlinear minimization techniques can then be used to further improve the results. This approach has the following advantages over the traditional "*motion from stereo*" approach:

- Each camera can be calibrated independently, and thus the calibration stage is easier and superior calibration results could be obtained.
- The relative geometry of the cameras can be changed during action (e.g., convergence, fixation, etc). It should, of course, remain constant during the two snapshots in consideration.

Because of the exploitation of information redundancy, the approach gains over the traditional "*motion and structure from motion*" approach in that less features and less motions are required, and thus more robust estimation of motion and structure can be obtained. The reader is referred to [28] for more details.

3.2 Uncalibrated Stereo

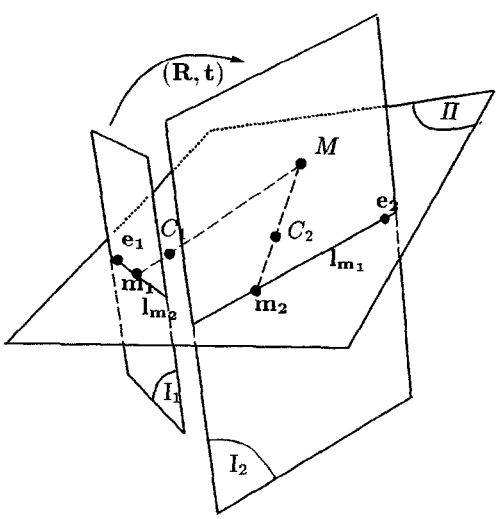


Fig. 5. The epipolar geometry

We consider here a more general problem: neither the intrinsic parameters nor the extrinsic parameters of the stereo system are known. Without knowledge of these, we can determine the epipolar geometry of the stereo rig from two images, which is summarized by a 3×3 matrix called the *fundamental matrix* if a camera is modeled as a pinhole. A stereo rig for which the fundamental matrix $\mathbf{F}$ is known is said to *be weakly calibrated.*

Epipolar geometry and fundamental matrix. Consider the case of two cameras as shown in Fig. 5. Let C_1 and C_2 be the optical centers of the first and second cameras, respectively. Given a point $\mathbf{m_1}$ in the first image, its corresponding point in the second image is constrained to lie on a line called the *epipolar line* of $\mathbf{m_1}$, denoted by $\mathbf{l_{m_1}}$. The line $\mathbf{l_{m_1}}$ is the intersection of the plane Π, defined by $\mathbf{m_1}$, C_1 and C_2 (known as the *epipolar plane*), with the second image plane I_2. Furthermore, all epipolar lines of the points in the first image pass through a common point $\mathbf{e_2}$, which is called the *epipole*. $\mathbf{e_2}$ is the intersection of the line C_1C_2 with the image plane I_2. Finally, one can easily see the symmetry of the epipolar geometry. The corresponding point in the first image of each point $\mathbf{m_2}$ lying on $\mathbf{l_{m_1}}$ must lie on the epipolar line $\mathbf{l_{m_2}}$, which is the intersection of the same plane Π with the

first image plane I_1. The symmetry leads to the following observation. If $\mathbf{m_1}$ (a point in I_1) and $\mathbf{m_2}$ (a point in I_2) correspond to a single physical point M in space, then $\mathbf{m_1}$, $\mathbf{m_2}$, C_1 and C_2 must lie in a single plane.

Let the displacement from the first camera to the second be $(\mathbf{R}, \mathbf{t})$. Let $\mathbf{m_1}$ and $\mathbf{m_2}$ be the images of a 3-D point M on the cameras. Without loss of generality, we assume that M is expressed in the coordinate frame of the first camera. Under the pinhole model, we have the following two equations:

$$s_1\tilde{\mathbf{m}}_1 = \mathbf{A_1}\,[\mathbf{I}\ \mathbf{0}]\,\widetilde{\mathbf{M}} \qquad \text{and} \qquad \mathbf{s_2}\tilde{\mathbf{m}}_2 = \mathbf{A_2}\,[\mathbf{R}\ \mathbf{t}]\,\widetilde{\mathbf{M}}\,,$$

where $\tilde{}$ indicates the homogeneous coordinates, $\mathbf{A_1}$ and $\mathbf{A_2}$ are the intrinsic matrices of the first and second cameras, respectively. Eliminating M, s_1 and s_2 from the above equations, we obtain the following fundamental equation

$$\tilde{\mathbf{m}}_2^T\mathbf{A_2^{-T}}[\mathbf{t}]_\times\mathbf{R}\mathbf{A_1^{-1}}\tilde{\mathbf{m}}_1 = \mathbf{0}\,, \tag{1}$$

where $[\mathbf{t}]_\times$ is an antisymmetric matrix defined by $\mathbf{t}$ such that $[\mathbf{t}]_\times\mathbf{x} = \mathbf{t}\times\mathbf{x}$ for all 3-D vector $\mathbf{x}$ ($\times$ denotes the cross product).

Equation (1) is a fundamental constraint underlying any two images if they are perspective projections of one and the same scene. Let $\mathbf{F} = \mathbf{A_2^{-T}}[\mathbf{t}]_\times\mathbf{R}\mathbf{A_1^{-1}}$, $\mathbf{F}$ is known as the fundamental matrix of the two images. Without considering 3-D metric entities, we can think of the fundamental matrix as providing the two epipoles (i.e., the vertexes of the two pencils of epipolar lines) and the 3 parameters of the homography between these two pencils. This is the only geometric information available from two uncalibrated images [16, 15]. This implies that the fundamental matrix has only seven degrees of freedom. Indeed, it is only defined up to a scale factor and its determinant is zero. Geometrically, $\mathbf{F}\tilde{\mathbf{m}}_1$ defines the epipolar line of point $\mathbf{m_1}$ in the second image. Equation (1), i.e. $\boxed{\tilde{\mathbf{m}}_2^T\mathbf{F}\tilde{\mathbf{m}}_1 = \mathbf{0}}$, says no more than that the correspondence in the right image of point $\mathbf{m_1}$ lies on its epipolar line. Transposing (1) yields the symmetric relation from the second image to the first image.

It can be shown that the fundamental matrix $\mathbf{F}$ is related to the essential matrix $\mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$ by $\mathbf{F} = \mathbf{A_2^{-T}}\mathbf{E}\mathbf{A_1^{-1}}$. It is thus clear that if the camera intrinsic parameters are known, the problem then becomes that described in the last subsection.

Image matching and fundamental matrix estimation. If we are given at least 7 point matches in a general position, the fundamental matrix $\mathbf{F}$ can be determined. However, matching different images remains one of the bottlenecks in computer vision. We have developed a robust technique which exploits the only *but unknown* geometric constraint, i.e., the epipolar constraint, to establish robustly correspondences between two perspective images of a single scene [31]. However, in order to reduce the complexity of the algorithm, we first extract high curvature points and then match them using a classical correlation technique followed by a fuzzy relaxation procedure. Because only heuristics have been used so far, many matches are possibly false. If we use directly this set of matches to estimate the fundamental matrix using a least-squares technique, the false matches will completely spoil the estimation process, and the final estimate of the fundamental matrix will be useless. This is why we use a robust technique—the Least

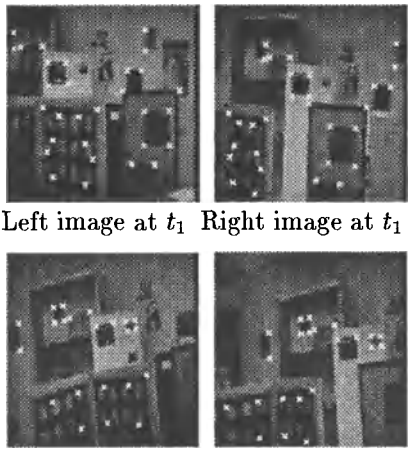
Median of Squares (LMedS)—to discard false matches in this set. The epipolar geometry can then be accurately estimated using a meaningful image criterion. More matches are eventually found, as in stereo matching, by using the recovered epipolar geometry. A large number of experiments have been carried out, and very good results have been obtained. More details can be found in [31], and the executable code is available through anonymous `ftp` on `krakatoa.inria.fr` under the name `pub/robotvis/BINAIRES/sun4OS4/image-matching.gz`

Projective reconstruction. It is shown that, in the case of a binocular stereo rig, the projective structure of the world can be recovered as soon as the epipolar geometry of the stereo rig, i.e. the fundamental matrix, is known. If more information, such as the determination of a plane at infinity, is added, the affine structure can then be recovered. It is argued that for artificial systems, such as robots, it is sometimes sufficient to think of physical space as being embedded in an affine or even a projective space. The reader is referred to [5] for more discussions. In [21], it is shown that the 3-D convex hull of an object can be computed from a weakly calibrated stereo pair.

4 Self-Calibration From Motion of a Stereo Rig

As said in the introduction, it is impossible to calibrate cameras on line with the classical techniques, while they are involved in a visual task. Any change of camera calibration occuring during the performance of the task cannot be corrected without interrupting the task. The change may be deliberate, for example the focal length of a camera may be adjusted, or it may be accidental, for example the cameras may undergo small mechanical or thermal changes. Recently, a number of researchers in Computer Vision and Robotics are trying to develop online camera calibration techniques, known as *self-calibration.* The idea is to calibrate a camera by just moving it in the surrounding environment. Assuming that the motion is rigid, then the epipolar constraint, in terms of the fundamental matrix, provides two constraints on the camera intrinsic parameters. From the definition of $\mathbf{F}$, we have $\mathbf{E} = \mathbf{A}^{\mathbf{T}}\mathbf{F}\mathbf{A}$, where $\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}$ is the essential matrix. As is well known [12], $\mathbf{E}$ is subject to two independent polynomial constraints in addition to $\det \mathbf{E} = 0$. This implies that the entries of $\mathbf{A}$ are subject to two independent polynomial constraints inherited from $\mathbf{E}$. Alternatively, Maybank and Faugeras [16] derive two so-called *Kruppa equations* which constrain the intrinsic parameters by observing the absolute conic which is invariant under the rigid motions of space. Hartley [10] proposed a singular-value-decomposition method to compute the focal lengths from a pair of images if all other cameras parameters are known. Maybank and Faugeras [16] proposed a theory of self-calibration. They showed that a camera can be in general completely calibrated from three different displacements. At the same time, they proposed an algorithm using tools from algebraic geometry. However, the algorithm is very sensitive to noise, and is of no practical use. Luong, in cooperation with them, has developed a real practical system as long as the points of interest can be located with sub-pixel precision, say 0.2 pixels, in image planes [15, 6].

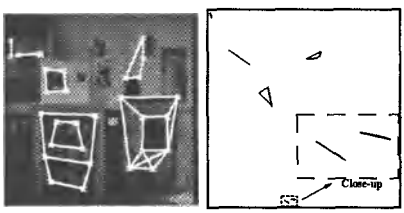
In [37], we have developed a self-calibration method for a binocular stereo rig from one displacement using a simplified camera model (the principal points are assumed to be known). We have made this simplification because it is very difficult to estimate precisely the position of the principal point by calibration, and is in practice very close to the image center. This has been shown experimentally. The formulation, however, can be easily extended to include all parameters. The problem of self-calibration and metric reconstruction (up to a scale factor) is formulated as a problem of estimating unknowns such that the discrepancy from the epipolar constraint, in terms of sum of squared distances between points and their corresponding epipolar lines, is minimized. Because of the exploitation of information redundancy in the stereo system, our approach yields more robust calibration results than those which consider a single camera.



Left image at t_1 Right image at t_1

Left image at t_2 Right image at t_2

Fig. 6. Images with overlay of the points of interest used for self-calibration



(a) (b)

Fig. 7. Final 3-D reconstruction result. (a): Back projection on the left image at t_1; (b): Projection on a plane perpendicular to the image plane (top view)

Figure 6 shows two pairs of stereo images used in one of our experiments. The points of interest used for self-calibration are also shown as indicated by the white crosses. Our proposed technique first estimates the intrinsic and extrinsic parameters of the cameras, and then performs a metric reconstruction (up to a scale factor). Figure 7a shows the back projection of the reconstructed 3-D points on the left image at t_1, which are artificially linked by white line segments, together with the original 2-D points as indicated by white crosses. The projected and original points coincide very well as shown in Fig. 7a. Figure 7b shows the projection of the 3-D reconstruction on a plane perpendicular to the image plane, which is expected to be parallel to the ground plane. A closeup of the foreground is also given. We see clearly that the reconstructed points of the foreground lie in two planes. The reconstruction of the background is however noisier due to the poor location of the 2-D points. To give an idea of the quantitative performance, we compute the scalar factor, the rotation matrix and the translation between the points reconstructed and those measured manually. We then apply the estimated transformation to the reconstructed points, and finally compute the distances

between the transformed points and the manually measured ones. The error (the root of the mean squares of the distances) is found to be 0.86 millimeters (the grid size is about 300 millimeters).

References

1. N. Ayache. *Artificial Vision for Mobile Robots.* MIT Press, 1991.
2. H. Chen and T. Huang. Maximal matching of 3-D points for multiple-object motion estimation. *Pattern Recog.*, 21(2):75–90, 1988.
3. H. Chen and T. Huang. Matching 3-D line segments with applications to multiple-object motion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10):1002–1008, 1990.
4. R. Deriche and O. Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, Nov. 1990.
5. O. Faugeras. Stratification of 3-d vision: projective, affine, and metric representations. *J. of the Optical Society of America A*, 12(3):465–484, Mar. 1995.
6. O. Faugeras, T. Luong, and S. Maybank. Camera self-calibration: theory and experiments. In Sandini [22], pages 321–334.
7. O. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, FL, June 1986. IEEE.
8. P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1), Winter 1993.
9. W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *Int'l J. of Robotics Research*, 3(3):3–35, 1984.
10. R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In Sandini [22], pages 579–587.
11. B. Hotz, Z. Zhang, and P. Fua. Incremental construction of local DEM for an autonomous planetary rover. In *Proc. Workshop on Computer Vision for Space Applications*, pages 33–43, Antibes, France, Sept. 1993.
12. T. S. Huang and O. D. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, Dec. 1989.
13. M. Jenkin and J. Tsotsos. Applying temporal constraints to the dynamic stereo problem. *Computer Vision, Graphics, and Image Processing*, 24:16–32, 1986.
14. Y. Kim and J. Aggarwal. Determining object motion in a sequence of stereo images. *IEEE J. RA*, 3(6):599–614, Dec. 1987.
15. Q.-T. Luong. *Matrice Fondamentale et Calibration Visuelle sur l'Environnement-Vers une plus grande autonomie des systèmes robotiques.* PhD thesis, Université de Paris-Sud, Centre d'Orsay, Dec. 1992.
16. S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *Int'l J. of Computer Vision*, 8(2):123–152, Aug. 1992.
17. E. E. Milios. Shape matching using curvature processes. *Computer Vision, Graphics, and Image Processing*, 47:203–226, 1989.
18. N. Navab and Z. Zhang. From multiple objects motion analysis to behavior-based object recognition. In *Proc. ECAI 92*, pages 790–794, Vienna, Austria, Aug. 1992.
19. N. Navab, Z. Zhang, and O. Faugeras. Tracking, Motion and Stereo: A robust and dynamic cooperation. In *7th Scandinavian Conf. Image Analysis*, pages 98–105, Aalborg University, Denmark, Aug. 1991.

20. S. Pollard, J. Porrill, J. Mayhew, and J. Frisby. Matching geometrical descriptions in three-space. *Image and Vision Computing*, 5(2):73–78, 1987.
21. L. Robert and O. Faugeras. Relative 3d positioning and 3d convex hull computation from a weakly calibrated stereo pair. In *Proc. 4th Int'l Conf. Computer Vision*, pages 540–544, Berlin, Germany, May 1993.
22. G. Sandini, editor. *Proc. 2nd European Conf. Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
23. I. Sethi and R. Jain. Finding trajectories of Feature Points in a Monocular Image Sequence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(1):56–73, Jan. 1987.
24. R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. of Robotics and Automation*, 3(4):323–344, Aug. 1987.
25. A. van Doorn and J. Koenderink. Spatiotemporal integration in the detection of coherent motion. *Vision Research*, 24(1):47–53, 1984.
26. D. Walters. Selection of image primitives for general-purpose visual processing. *Computer Vision, Graphics, and Image Processing*, 37(3):261–298, 1987.
27. J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.
28. Z. Zhang. Motion and structure of four points from one motion of a stereo rig with unknown extrinsic parameters. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 556–561, New York, June 1993. Full version to appear in IEEE Trans. PAMI.
29. Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int'l J. of Computer Vision*, Vol.13, No.2, pages 119–152, 1994.
30. Z. Zhang. Token tracking in a cluttered scene. *Image and Vision Computing*, 12(2):110–120, Mar. 1994.
31. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 1994. to appear. Also INRIA Research Report No.2273, May 1994.
32. Z. Zhang and O. Faugeras. A 3-D World Model Builder with a Mobile Robot. *Int'l J. of Robotics Research*, 11(4):269–285, Aug. 1992.
33. Z. Zhang and O. Faugeras. Estimation of displacements from two 3D frames obtained from stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(12):1141–1156, Dec. 1992.
34. Z. Zhang and O. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *Int'l J. of Computer Vision*, 7(3):211–241, 1992.
35. Z. Zhang and O. Faugeras. Finding planes and clusters of objects from 3D line segments with application to 3D motion determination. *CVGIP: Image Understanding*, 60(3):267–284, Nov. 1994.
36. Z. Zhang and O. D. Faugeras. *3D Dynamic Scene Analysis: A Stereo Based Approach*. Springer, Berlin, Heidelberg, 1992.
37. Z. Zhang, Q.-T. Luong, and O. Faugeras. Motion of an uncalibrated stereo rig: self-calibration and metric reconstruction. In *Proc. Int'l Conf. Pattern Recognition*, volume I, pages 695–697, Jerusalem, Israel, Oct. 1994. Full version to appear in IEEE Trans. Robotics and Automation.

Color Stereo Vision: Use of Appearance Constraint and Epipolar Geometry for Feature Matching *

Ming Xie and Lai Yuan Liu

School of Mechanical & Production Engineering
Nanyang Technological University, Singapore 2263
email: mmxie@ntuix.ntu.ac.sg

Abstract. The difficult problem of stereo vision is how to establish correspondence between features extracted from a pair of images. The difficulty is due to ambiguities or inconsistence of available information in images. In this paper, we investigate stereo correspondence problem in the framework of color stereo vision. We propose the use of an appearance constraint and generalized epipolar geometry to develop a matching algorithm.

1 Introduction

Stereo or motion correspondence is a very important topic in computer vision. It provides the necessary input to modules dedicated to 3D reconstruction, motion estimation and possibly on-line calibration. The correspondence problem in stereo vision or motion analysis can be shortly stated as how to identify, from two (stereo or consecutive) images, a pair of geometrical primitives (including image points) which are the projection of a same physical primitive.

Stereo or motion correspondence problem has been widely investigated in the past. This has been reflected by a large number of papers published in this area ([1] - [22]). The algorithms developed so far mainly fall into the same framework of "similarity-based matching". For instance, there are methods using: (a) graph or structural similarity ([1], [4], [11], [12], [13]), (b) shape similarity ([2], [3], [5], [6], [7], [17], [20]) and (c) intensity similarity ([15], [10], [9], [16], [19], [21]). Stereo or motion correspondence is a very difficult problem because it is ill-conditioned in general case. The difficulty of solving stereo or motion correspondence problem is largely due to the impossibility of finding any constraint which must be not only a necessary but also sufficient condition that stereo or motion correspondence has to satisfy. Because of this, it is inevitable that there are ambiguities in the results of established matches. In order to reduce or eliminate the ambiguity, efforts have been made to find more distinctive measures (e.g., projective invariant) and also to introduce physical constraints such as

* This work is supported by ARC1/94 project funded by the Ministry of Education (Singapore)

epipolar line, continuity in disparity gradients, figural consistency, ordering constraint, Hough Transform, etc. Although promising results have been achieved in stereo (or motion) correspondence, improvement is still possible. The aim of this paper is to investigate the use of color information together with epipolar geometry to develop a matching algorithm which has less ambiguity in established matches.

2 Methodology

The basic idea underlying most matching algorithms is as follows: First to establish the correspondence of features by using a similarity function/constraint, and then to eliminate ambiguities in established matches by using constraints such as continuity in disparity gradients, figural consistency, ordering constraint, Hough Transform, etc. This two-step procedure can be applied sequentially in one pass ([9], [21], [22]) or iteratively in few passes ([10], [16]). The success of a matching algorithm depends on the performance of each of the above two steps. The most critical one is the first step. So far, information on intensity, edgeness, texture, shape and geometrical structure have been used to test similarity among features by researchers [19].

In [14], a detailed discussion on usefulness of color information can be found. Here, we further investigate the use of color information to perform stereo correspondence. The advantages of using color information coded on R-G-B space can be understood by the following facts:

- Consistence of Color Information:
  A color can be represented by the combination of three primary color components (R(red), G(green), B(blue)). One combination of (R, G, B) corresponds to only one physical color. There is no ambiguity. This is not true for intensity information.
- Appearance Constraint:
  In stereo vision, we assume that a pair of images are captured at the same time under the same lighting condition with two cameras of the same characteristics. Suppose that the baseline of two cameras is negligible compared to the distance between object and vision sensor. Based on this assumption, two images of a physical primitive observed by stereo cameras should have a same color. We call this condition as *appearance constraint* (or color consistency constraint).
- Good Tokens Detected From Color:
  Feature detection with color will have higher SNR (Signal/Noise Ratio) than with intensity information. For example, the edge strength between a color region of (R=100, G=50; B=30) and a color region of (R=40, G=50; B=30) is 60 (pixels). If we use only intensity information, the edge strength will only be 20 (pixels) ($20 = (100 + 50 + 30)/3 - (40 + 50 + 30)/3$). Moreover, color edges are more than intensity edges. For instance, the boundary between a color region of (R=200, G=10; B=100) and a color region of (R=100,

G=200; B=10) will not be detectable if only intensity information is being used.

3 Use of Color Edge As Primitives

Color edge detection has been investigated in the past. A survey can be found in [14]. In the area of intensity edge detection, a lot of algorithms exist. The most representative one is Canny-Deriche edge detector which operates in four steps:

1. Compute X gradient map after smoothing in Y direction.
2. Compute Y gradient map after smoothing in X direction.
3. Non-Maxima suppression.
4. Threshold by hysteresis.

As mentioned above, intensity edges are only a subset of color edges. Furthermore, SNR of color edge is higher than SNR of intensity edge. In order to preserve the useful information encoded in color, we propose the following Canny-Deriche edge detector of color version:

1. For each component image R, G or B, compute X gradient map after smoothing in Y direction. We denote (R_x, G_x, B_x) the three X gradient map corresponding to the component images R, G and B, respectively.
2. For each component image R, G or B, compute Y gradient map after smoothing in X direction. We denote (R_y, G_y, B_y) the three Y gradient map corresponding to the component images R, G and B, respectively.
3. Compute the combined gradient map as follows:

$$\sqrt{(R_x + G_x + B_x)^2 + (R_y + G_y + B_y)^2}.$$

4. Non-Maxima suppression.
5. Threshold by hysteresis.

One may easily notice that the computation of color edges is much more time consuming than the computation of intensity edge. In some applications where only higher SNR is expected, one can use the sum (R+G+B) as input image to the Canny-Deriche edge detector of monochrome version. This will still produce better results than just working with intensity image.

4 Stereo Matching By Using Appearance Constraint

In monochrome stereo vision, one common approach is to use cross-correlation to first establish feature correspondences. The standard form of cross-correlation in stereo matching is as follows:

$$C(u_l, v_l, u_r, v_r) = \frac{1}{K} \bullet \frac{\sum_{x=-n}^{n} \sum_{y=-m}^{m} F(x, y, u_l, v_l, u_r, v_r)}{\sigma_1 \, \sigma_2} \tag{1}$$

where:

$$\begin{cases} F(x, y, u_l, v_l, u_r, v_r) \\ = [I_l(x + u_l, y + v_l) - \bar{I}_l(u_l, v_l)][I_r(x + u_r, y + v_r) - \bar{I}_r(u_r, v_r)] \\ \sigma_1 = \sqrt{\sum_{x=-n}^{n} \sum_{y=-m}^{m} (I_l(x + u_l, y + v_l) - \bar{I}_l(u_l, v_l))^2} \\ \sigma_2 = \sqrt{\sum_{x=-n}^{n} \sum_{y=-m}^{m} (I_r(x + u_r, y + v_r) - \bar{I}_r(u_r, v_r))^2} \\ K = (2\,n + 1)^2 \bullet (2\,m + 1)^2 \end{cases} \tag{2}$$

And, $I_l(.)$ and $I_r(.)$ are intensity functions of left and right images in stereo vision. $(\bar{I}_l, \bar{I}_r)$ are the mean values of $I_l(.)$ and $I_r(.)$ in the window $(2\,n+1) \times (2\,m+1)$ centered at pixels (u_l, v_l) and (u_r, v_r), respectively. (x, y) are coordinates of image pixels.

From Eq.1, we can see the following two problems:

- Cross-correlation requires mean subtraction for functions $I_l(.)$ and $I_r(.)$. In this way, it is only sensitive to the shapes of the functions $I_l(.)$ and $I_r(.)$, but not to their mean values. This property is not desirable for color images because absolute values in (R, G, B) encode useful information.
- Intensity-based matching relies on intensity similarity. This is a very weak constraint because a same intensity value may correspond to a wide range of different colors. For instance, the intensity value 100 can be the combination of the color (R=10, B=20, G=70) or (R=20, B=20, G=60), and so forth. This explains why intensity-based matching algorithms greatly suffer the problem of ambiguity in established matches.

Cross-correlation-based stereo matching algorithms implicitly uses the following appearance constraint:

> **Appearance Constraint:** In stereo vision, a pair of images are captured at the same time under the same lighting condition with two cameras of the same characteristics. Suppose that the distance between two cameras is negligible compared to the distance between object and vision sensor. Under these conditions, two images of a physical primitive observed by stereo vision should have a same color (or intensity value).

This constraint only constitutes a necessary condition that a pair of matching candidates must meet. It is definitely not a sufficient condition. Intrinsically, any matching algorithms based on the use of non-sufficient conditions have ambiguity in established matches. The causes of matching ambiguity are basically two:

- Scene-Caused Ambiguity:
  This is due to the presence of similar (sometime periodic) pattern of color and/or intensity in a physical scene. In this case, it is very difficult to identify correct matches because matching operation is basically performed locally in images.
- Sensor/algorithm-Caused Ambiguity:
  This is due to the limitation of sensor (for instance, monochrome camera

produces a same output for two color regions (R=100, G=150, B=200) and (R=200, G=100, B=150)), similarity function used by a matching algorithm, or the combination of these two causes.

In our study, we also use Appearance Constraint. Since a color encoded in R-G-B space is unique, we therefore propose a color distance based similarity function, that is:

$$C(u_l, v_l, u_r, v_r) = 1 - \frac{\sum_{x=-n}^{n} \sum_{y=-m}^{m} (A^2 + B^2 + C^2)}{k \bullet (2\,n+1) \bullet (2\,m+1)}. \tag{3}$$

where:

$$\begin{cases} k = 3 \bullet 255^2 \\ A = R_l(x+u_l, y+v_l) - R_r(x+u_r, y+v_r) \\ B = G_l(x+u_l, y+v_l) - G_r(x+u_r, y+v_r) \\ C = B_l(x+u_l, y+v_l) - B_r(x+u_r, y+v_r) \end{cases} \tag{4}$$

and, $R_{l,r}(.)$, $G_{l,r}(.)$ and $B_{l,r}(.)$ are image functions corresponding to red, green and blue component images, respectively. Suffixes (l, r) denote left or right camera.

The similarity function defined in Eq.3 varies in the range [0,1]. Now, stereo matching can be summarized as follows: Given a pixel (u_l, v_l) in left image, find a pixel $(\hat{u}_r, \hat{v}_r)$ in right image which maximizes the similarity function in Eq.3:

$$C(u_l, v_l, \hat{u}_r, \hat{v}_r) = \max_{(u_r, v_r) \in W} \{\, C(u_l, v_l, u_r, v_r) \,\}. \tag{5}$$

where "W" denote searching window in right image.

The computation of $C(u_l, v_l, u_r, v_r)$ is time consuming because each pixel involves three multiplications. In practice, a good approximation is to use the following similarity functions:

$$C(u_l, v_l, u_r, v_r) = [C_R(.) + C_G(.) + C_B(.)]/3 \tag{6}$$

where:

$$\begin{cases} C_R(.) = 1 - \frac{\sum_{x=-n}^{n} \sum_{y=-m}^{m} (\|R_l(x+u_l, y+v_l) - R_r(x+u_r, y+v_r)\|)}{255 \bullet (2\,n+1) \bullet (2\,m+1)} \\ C_G(.) = 1 - \frac{\sum_{x=-n}^{n} \sum_{y=-m}^{m} (\|G_l(x+u_l, y+v_l) - G_r(x+u_r, y+v_r)\|)}{255 \bullet (2\,n+1) \bullet (2\,m+1)} \\ C_B(.) = 1 - \frac{\sum_{x=-n}^{n} \sum_{y=-m}^{m} (\|B_l(x+u_l, y+v_l) - B_r(x+u_r, y+v_r)\|)}{255 \bullet (2\,n+1) \bullet (2\,m+1)} \end{cases} \tag{7}$$

The similarity function C(.) in Eq.6 also takes values in the range [0,1]. A very interesting advantage of doing so is the possibility of qualitatively classifying the matches into four classes:

- Class I: For each confirmed match (u_r, v_r) where $C(.)$ reaches a maximum value, $C_R(.)$, $C_G(.)$ and $C_B(.)$ all take a maximum value also.
- Class II: For each confirmed match (u_r, v_r) where $C(.)$ reaches a maximum value, only two of $C_R(.)$, $C_G(.)$ and $C_B(.)$ take a maximum value.
- Class III: For each confirmed match (u_r, v_r) where $C(.)$ reaches a maximum value, only one of $C_R(.)$, $C_G(.)$ or $C_B(.)$ takes a maximum value.
- Class IV: For each confirmed match (u_r, v_r) where $C(.)$ reaches a maximum value, none of $C_R(.)$, $C_G(.)$ and $C_B(.)$ takes a maximum value.

5 Generalized Epipolar Geometry

It can be seen that the similarity function defined in Eq.6 mainly involves addition/subtractions. Its computational efficiency also depends on searching window "W". In order to limit the size of searching window, we must introduce additional constraint. In stereo vision, one very powerful constraint is *epipolar geometry.*

It is well known that the following equation describes the epipolar geometry in stereo vision:

$$(u_l, v_l, 1) \bullet F \bullet \begin{pmatrix} u_r \\ v_r \\ 1 \end{pmatrix} = 0. \tag{8}$$

where $F = \{f_{i,j},\ i = 1,2,3;\ j = 1,2,3\}$ is a 3×3 fundmental matrix with rank of 2 [8].

By considering $f_{3,3}$ to be a constant, Eq.8 can be simplified to:

$$(u_l, v_l, 1) \bullet H \bullet \begin{pmatrix} u_r \\ v_r \\ 1 \end{pmatrix} = 0. \tag{9}$$

with:

$$H = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1.0 \end{bmatrix}. \tag{10}$$

We call Eq.9 *generalized epipolar geometry* because the eight unknowns (a, b, c, d, e, f, g, h) can be estimated from a linear system by using standard Least-Squares estimation method if a set of eight pairs of matched points are given.

For a pixel (u_l, v_l) in left image, Eq.9 describes an epipolar line in right image. The match of (u_l, v_l) must locate on this epipolar line. Therefore, the searching window can be specified according to generalized epipolar geometry.

6 Implementation and Results

The color stereo matching algorithm has been implemented on SUN Sparc 20 workstation. In our experiments, we set $n = m = 30$ (pixels). The resolution of images is 512×512. For each stereo pair of images, we first establish eight good matches from randomly selected points. At this stage, each match must fall into the quality category "Class I" of matching. From the matched eight pairs of points, we estimate the generalized epipolar geometry. Subsequently, we establish the stereo correspondence of a set of selected points. Here, we give some examples which are illustrated in Fig. 1, Fig. 2 and Fig. 3.

In terme of time performance, the matching of eight points will take about 4 minutes (without the use of epipolar geometry). By using the estimated epipolar geometry, the matching of one pair of points will take less than 1 second.

7 Conclusion

Color encoded in R-G-B space is more consistent than pure intensity information. The benefits of using color are two-fold: (a) the improvement of the quality in image feature extraction and (b) the increase of robustness in stereo matching. The only drawback of using color is the cost in computation due to the increased amount of image data. But, this problem can be overcome in practice by an effective combination of color components and an appropriate design of vision algorithms.

References

[1] Ayache, N and Faverjon, B.: Efficient registration of stereo images by matching graph descriptions of edge segments. International Journal of Computer Vision 1, (1987), 107-131.

[2] Bachelder, I. A. and Ullman. S.: Contour matching using local affine transformations. IEEE International Conference on Computer Vision and Pattern Recognition, (1993). 798-801.

[3] Brint, A. T. and Brady. M.: Stereo matching of curves. Journal of Image and Vision Computing 8, (1990), 50-56.

[4] Cohen, L. Vinet, L. Sander, P. T. and Gagalowicz, A.: Hierarchical region based stereo matching. IEEE International Conference on Computer Vision and Pattern Recognition, (1989), 416-421.

[5] Costabile, M. F. Guerra, C. and Pieroni. G. G.: Matching shapes: a case study in time-varying images. Computer Vision, Graphics and Image Processing 29, (1985), 296-310.

[6] Deriche, R and Faugeras. O.: 2-D curve matching using high curvature points: Application to stereo vision. International Conference on Pattern Recognition, (1990). 240-242.

[7] Ellis, T. Abbood, A. and Brillault. B.: Ellipse detection and matching with uncertainty. Image and Vision Computing 10, (1992), 271-276.

[8] Faugeras, O.: Three Dimensional Vision: A Geometrical Viewpoint. MIT Press, (1993), 189-199.

[9] Fua, P.: A parallel stereo algorithm that produces dense depth maps and preserves image features. Machine Vision and Applications 6, (1993), 35-49.

[10] Grimson, W. E. L.: Computational experiments with a feature based stereo algorithm. IEEE Trans. on PAMI 7, (1985), 17-34.

[11] Grosky, W. I. and Jain. R.: A pyramid-based approach to segmentation applied to region matching. IEEE Trans. on PAMI 8, (1986), 639-650.

[12] Horaud, R. and Skordas, T.: Structural matching for stereo vision. International Conference on Pattern Recognition, (1988), 439-445.

[13] Limozin, P. L. and Giraudon, G.: Stereo matching using contextual line region primitives. International Conference on Pattern Recognition, (1986). 974-977.

[14] Luong, Q. T.: Color In Computer Vision. in Handbook of Pattern Recognition and Computer Vision (Eds. C.H. Chen, L. F. Pau, P. S. P. Wang), (1993), 311-368.

[15] Marr, D. and Poggio, T.: Cooperative Computation of Stereo Disparity. Science 194, (1976)

[16] Pollard, S. B. Porrill, J. Mayhew, J. E. W. and Frisby, J. P.: Disparity gradient, Lipschitz continuity, and computing binocular correspondences. CVGIP: Image Understanding, (1989), 197-214.

[17] Porrill, J and Pollard, S.: Curve matching and stereo calibration. Image and Vision Computing 9, (1991), 45-50.

[18] Sherman, D. and Peleg, S.: Stereo by incremental matching of contours. IEEE Trans. on PAMI 12, (1990), 1102-1107.

[19] Weng, J. Ahuja, N. and Huang, T. S.: Two-view matching. International Conference on Computer Vision, (1988), 64-73.

[20] Zhang, D. Z. Perspective invariant description of planar point set and their application in matching. International Conference on Pattern Recognition, (1986), 1266-1268.

[21] Zhang, Z. Deriche, R. Faugeras, O. and Luong, Q. T.: A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. International Conference on Automation, Robotics and Computer Vision, (1994), 1102-1106.

[22] XIE, M.: A Cooperative Strategy For The Matching of Multi-level Edge Primitives. Journal of Image and Vision Computing Vol.13(2), (1995), 89-99.

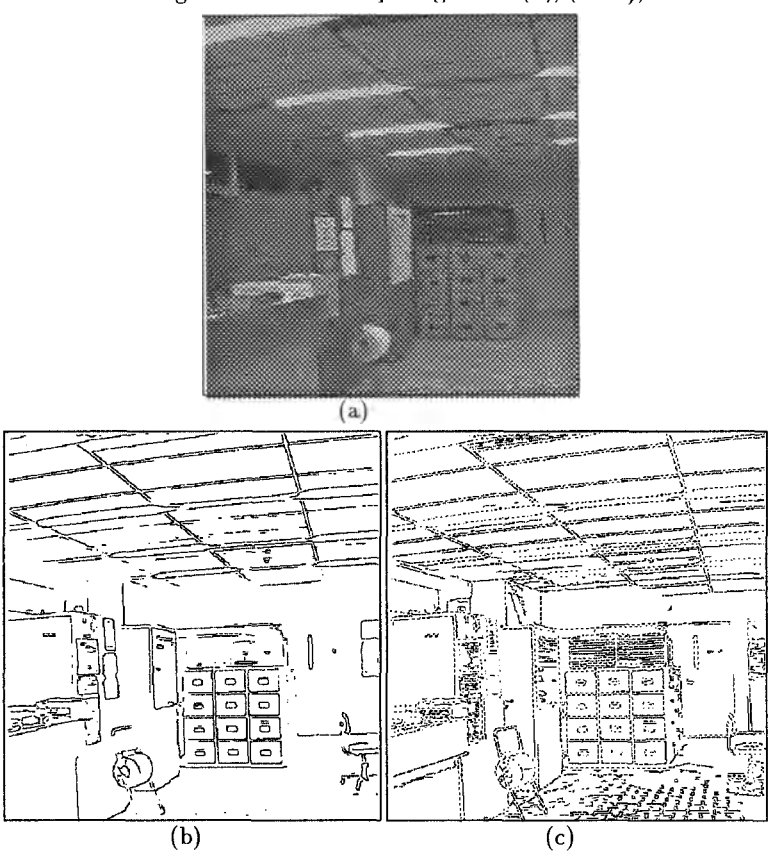
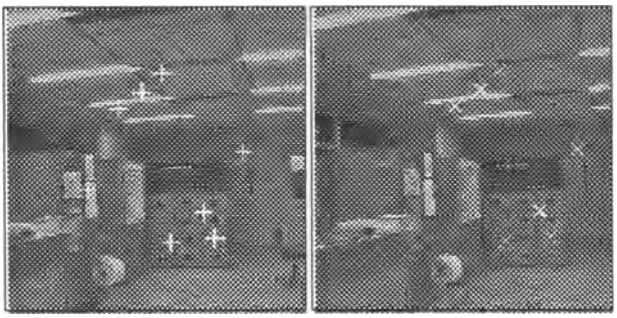
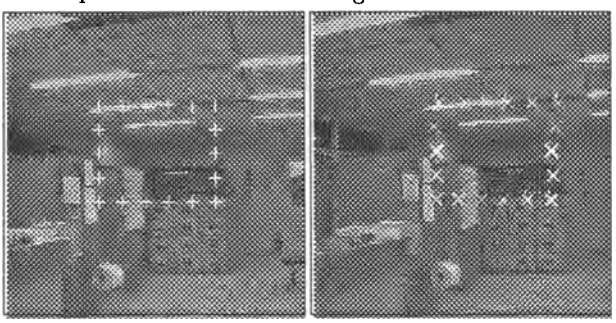


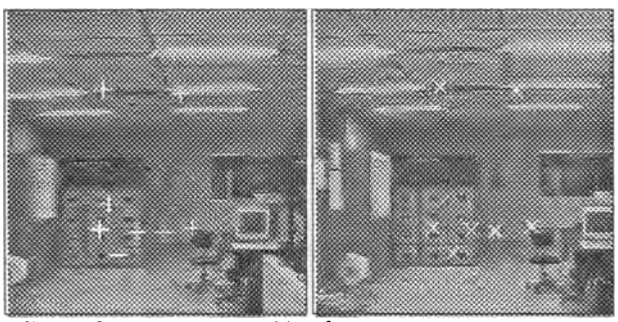
Fig. 1. Color Edge and Intensity Edge Comparison: (a) Original image; (b) Intensity edge and (c) Color edge.



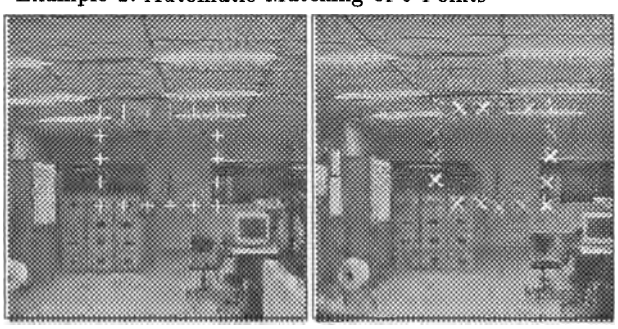
Example 1: Automatic Matching of 8 Points



Example 1: Matching With Epipolar Geometry

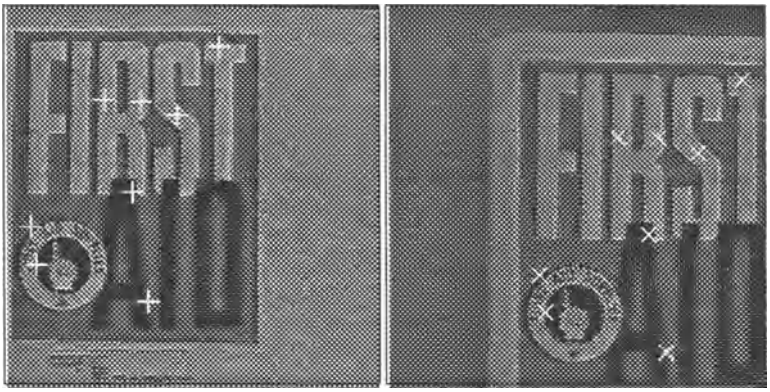


Example 2: Automatic Matching of 8 Points

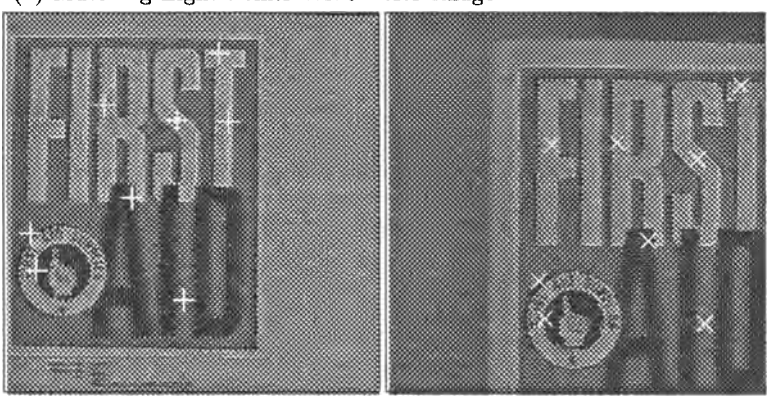


Example 2: Matching With Epipolar Geometry

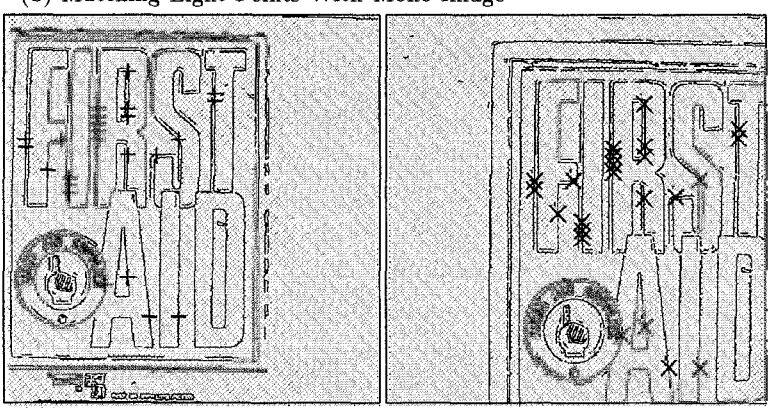
Fig. 2. Color Stereo Matching: For each example here, eight pairs of points are automatically matched in order to determine the epipolar geometry. Then, a set of selected points (forming a rectangle) are matched with the help of estimated epipolar geometry.



(a) Matching Eight Points With Color Image



(b) Matching Eight Points With Mono Image



(c) Matching Set of Points With Epipolar Geometry

Fig. 3. Color and Monochrome Stereo Matching Comparison: (a) Establish the matches of eight points using color image; (b) Establish the matches of eight points using monochrome image. (c) Match a set of points with the help of estimated epipolar geometry.

Unifying Stereo, Motion and Object Recognition via Epipolar Geometry

Gang Xu

Department of Systems Engineering, Osaka University
Toyonaka, Osaka 560, Japan
Email: xu@sys.es.osaka-u.ac.jp

Abstract. In this paper I try to show that through epipolar geometry we can unify the problems of image matching in stereo, motion and object recognition, which have been treated separately. Stereo matching has been known as a 1D search problem. But matching in motion and object recognition have been known as 2D search problems. I show that by recovering epipolar geometry underlying the images, the correspondence search problems in motion and object recognition can also be changed to be 1-dimensional, thus providing a framework to treat all these 3 problems in a unified way.

1 Introduction

Stereo, motion and object recognition are 3 main problems in computer vision [8]. All the 3 problems share the necessity of correspondence between views of completely or partially identical scene taken at different positions and/or at different times. Regardless of this commonality, the 3 problems have been treated in different frameworks.

In stereo, two images of the same scene are taken at moderately different positions. The two images are first matched and then the difference between the two images are used to recover the lost 3D dimension, depth [4]. Based on the assumption that the spatial relationship between the two cameras is fixed and known, almost all stereo algorithms employ the epipolar constraint to restrict the search area from the 2-dimensional image to the 1-dimensional epipolar lines [4].

In motion, since both the camera motion and object motion are arbitrary and instant, almost all motion correspondence algorithms set the search area to be a small 2D neighborhood. Due to the ambiguity involved in the 2D search, correspondence cannot be uniquely determined within each local area. This is called *aperture problem.* The conventioal wisdom is to impose the smoothness constraint to the whole 2D flow field so that we can obtain a solution for every (edge) point [6, 5].

In object recognition, the conventional wisdom of using 3D object models has been challenged. It has been shown that matched 2D views are equivalent to 3D data as object model for recognition [12]. Employing 2D views as object models, recognition becomes a problem of matching model views with input views. If

the matching is successful, then the objects in the input view and model views are the same; if not, they are different. Again, both matching between model views, and matching between model views and input view are a correspondence problem. There have been attempts to solve the correspondence problem using tracking[12, 13]. The limitation is obvious. Generally, the problem is yet to be solved.

In this paper I show that correspondence problems in both motion and object recognition can be reduced from a 2D search to a 1D search by first recovering the epipolar geometry underlying the images, thus providing a framework to treat image correspondence in stereo, motion and object recognition problems in a unified way. One of the advantages of this approach is that segmentation is simultaneous.

In the following, Section 2 gives an introduction to the epipolar geometry and its mathematical representations. Section 3 shows that by recovering the epipolar equation underlying images, the original 2D ambiguity of correspondence is reduced to 1D ambiguity, which is characterized by *disparity* defined along the epipolar lines. We can then represent the search space as $x - y - d$ space, or *spatial disparity space* as called in [15]. In Section 4, I redefine the problems of stereo, motion and object recognition within this unified framework. Section 5 proposes to recover the epipolar equations as clustering of local hypotheses. In Section 6 I show experimental results for stereo, motion, and object recognition based on the proposed approach.

2 Epipolar Geometry, Equation and Fundamental Matrix

An arbitrary motion can be represented by

$$\mathtt{X} = \mathbf{R}\mathtt{X}' + \mathbf{t}, \tag{1}$$

where $\mathtt{X}$ and $\mathtt{X}'$ are 3D coordinates before and after the motion, respectively, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector, respectively. Note that this formula is valid for any case of mere camera motion, mere object motion, and simultaneous camera and object motions. Given two images of the same rigid object, generally the corresponding points satisfy the following epipolar equation,

$$\tilde{\mathbf{x}}^T \mathbf{F} \tilde{\mathbf{x}}' = 0 \tag{2}$$

where $\tilde{\mathbf{x}} = [x, y, 1]^T$ and $\tilde{\mathbf{x}}' = [x', y', 1]^T$ are augmented vectors for the image coordinates of corresponding points, respectively, and $\mathbf{F}$ is the 3×3 *fundamental matrix*, whose components are functions of the rotation matrix, translation vector and camera's intrinsic parameters [3]. For linear approximations of perspective projection like weak perspective and paraperspetive projections [1], which are collectively called *affine cameras* [9], the upper left 4 components of $\mathbf{F}$ are zeros. Otherwise they are non-zeros for general perspective projection [10, 11, 14]. For affine cameras, the epipolar equation is linear about the image coordinates, much simpler than that for full perspective projection which is second-order

functions of the image coordinates. Since the coefficients are independent of the image coordinates, the epipolar lines are parallel everywhere. In the remaining part of this paper, we shall assume weak perspective projection if not mentioned otherwise, because of the simplicity and intuitive geometric interpretation of its epipolar equation.

3 Disparity and Spatial Disparity Space

In this section, we show how to define disparity from the epipolar equation under the weak perspective projection. For full perspective projection, see [14].

Expanding (2) for the weak perspective projection yields

$$f_{13}x + f_{23}y + f_{31}x' + f_{32}y' + f_{33} = 0, \tag{3}$$

which can be rewritten as

$$(-x\cos\alpha + y\sin\alpha) - \rho(-x'\cos\gamma + y'\sin\gamma) + \lambda = 0 \tag{4}$$

where

$$\begin{aligned} \alpha &= -\tan^{-1}\frac{f_{23}}{f_{13}}, & \gamma &= -\tan^{-1}\frac{f_{32}}{f_{31}}, \\ \rho &= \sqrt{\frac{f_{31}^2 + f_{32}^2}{f_{13}^2 + f_{23}^2}}, & \lambda &= \frac{f_{33}}{\sqrt{f_{13}^2 + f_{23}^2}}, \\ \theta &= \alpha - \theta, \end{aligned} \tag{5}$$

These parameters have a nice interpretation. α and γ are the first and third rotation angles in the Euler angle representation of 3D arbitrary rotation, ρ is the scale change between the two images due to the change in depth, λ is the translation along the direction perpendicular to the epipolar lines, and θ is the fronto-parallel rotation angle in KvD representation [10, 11, 7].

If we define a new coordinate system for each image as

$$\begin{aligned} \bar{x} &= x\cos\gamma + y\sin\gamma \\ \bar{y} &= -x\sin\gamma + y\cos\gamma \end{aligned} \tag{6}$$

and

$$\begin{aligned} \bar{x}' &= \rho(x'\cos\alpha + y'\sin\alpha) \\ \bar{y}' &= \rho(-x'\sin\alpha + y'\cos\alpha) - \lambda \end{aligned} \tag{7}$$

According to (4),

$$\bar{y} = \bar{y}' \tag{8}$$

Now disparity d can be defined as the difference between the new horizontal coordinates

$$d = \bar{x} - \bar{x}' \tag{9}$$

Once the epipolar geometry is known between two images, the positional difference between every pair of corresponding points can be characterized by the 1D disparity. Taking disparity as the 3rd dimension, we can construct $x-y-d$ space, or *Spatial Disparity Space* [15]. Looking for correspondence for point (x, y) is the same as looking for an active node $(x-y-d)$ in SDS. Looking for a disparity map corresponds to looking for a surface in SDS. The uniqueness constraint simply means that in SDS, each column (x, y) can have only one active point. The continuity constraint can be understood as a requirement that the active points of two neighbring columns should have disparity values with difference less than 1 pixel. Otherwise it means a discontinuity. The smoothness constraint can be defined as the differentials of this surface. With the help of SDS, matching can be easily visualized.

4 Redefining Stereo, Motion and Object Recognition

In this section, all the three problems are redefined identically as a two step process: epipolar geometry recovery and 1D search.

It is straightforward and conventional wisdom that stereo correspondence is a 1D search problem, if the epipolar geometry is known *a priori*. Here it suffices to only emphasize that the recovery of epipolar geometry is necessary if the two cameras are not carefully displaced in a special manner. If the scene is stationary, there is no difference between two motion images and a pair of stereo images. The epipolar geometry underlying the stereo images is the same as that underlying the two motion images caused by a moving camera. Thus, the algorithm to recover the epipolar geometry should also be the same. For how to recover epipolar geometry under full perspective projection for an uncalibrated stereo rig, see [2].

What differs between a general motion problem and a general stereo problem, is that in motion problem, we can have different motions simultaneously. The camera can move, the scene can move in a different way, and more importantly, there can be many objects moving independently in the scene, while in stereo, only the camera moves. Thus, between general motion images, there can be multiple epipolar geometries while in stereo there can only be one. If we can somehow recover all the epipolar geometries underlying two motion images, then the 2D search for correspondence is reduced from to a 1D search along multiple epipolar lines. The *aperture problem* no longer exists, though the ambiguity along epipolar lines still exists as in stereo. For multiple motions, we can just add to SDS more disparity layers corresponding to different motions. One of the advantages of this approach is that segmentation is simultaneous.

Object recognition here does not involve any explicit 3D models. Both the models and inputs are 2D images. Thus recognition is actually a judgement of existence of correspondence between model views and input views. There are two kinds of correspondence problems. One is correspondence between model views if multiple model views are used. This is completely the same problem as binocular stereo from the computational point of view. The other is between model

view(s) having only the model object and input view often having background objects. As in stereo and motion, if the object in one model view is the same as the object in the input view, then the feature points must satisfy the same epipolar equation, and feature points in model view(s) can be matched against feature points in the input view. The problem is then divided into one of finding possible epipolar geometries between a model view and the input view, and one of finding legitimate correspondence of image points based on the obtained epipolar geometries recovered. If matching is successful, then the two images are said to be the same object; otherwise, they are not. If the input view also includes other objects and background, finding the corresponding points to the model is also segmenting the object from other objects and background. The part of recovering epipolar geometry between a model view and an input view is essentially the same as motion images and stereo images, but usually there are other objects and background in the input view, which brings difficulty to epipolar equation recovery.

5 Recovery of Epipolar Equations by Clustering

To determine an epipolar equation, at least 4 pairs of point matched are needed. Since we do not know *a priori* which points belong to which group, grouping techniques are required. Given $n(n \geq 4)$ pairs of point matches $\mathbf{p}_i = [\, x_i, y_i, x_i', y_i' \,]^T$, $i = 1, ..., n$, $\mathbf{f} = [\, f_{13}, f_{23}, f_{31}, f_{32} \,]^T$ is determined as the eigen vector corresponding to the smallest eigenvalue of $\mathbf{W} = \sum_{i=1}^{n} (\mathbf{p}_i - \mathbf{p}_0)(\mathbf{p}_i - \mathbf{p}_0)^T$, and f_{33} is determined as $f_{33} = -\mathbf{p}_0^T \mathbf{f}$, where $\mathbf{p}_0 = \frac{1}{n} \sum_{i=1}^{n} \mathbf{p}_i$. For details, see [11, 14].

It is shown in [14] that, of the motion parameters in (5) scale change ρ and fronto-parallel rotation angle θ are much more stable than angle α for the 3D rotation axis and translation λ along the direction perpendicular to the epipolar lines. We choose to find groups in the space of motion parameters $\rho - \theta - \alpha - \lambda$ instead of the space of the coefficients in the epipolar equation.

The strategy for grouping is to determine an epipolar equation for each quadriple of point matches, which in turn determines a point in the motion parameter space, and to find clusters in that space, each of which is a group of point matches sharing the same epipolar equation.

Assume that in each cluster the distribution of the points obeys the Gaussian function with a mean $\bar{\mathbf{m}}$ and a covariance matrix Σ,

$$p(\mathbf{x}) = (2\pi)^{-2} \mid \Sigma \mid^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{m} - \bar{\mathbf{m}}_i)^T \Sigma^{-1} (\mathbf{m} - \bar{\mathbf{m}}_i)\}. \tag{10}$$

where the covariance matrix Σ is estimated using matched images, and the mean vectors $\bar{\mathbf{m}}_i$ are to be found. We employ an incremental clustering procedure. Each point in the space is initially assumed to be the mean of a cluster, and is moved incrementally to a position of higher concentration. The new position is computed using only those points within a certain distance updated for each new position. The process is repeated until the position has arrived to an equibrilium.

The squared Mahalanobis distance from $\mathbf{x}$ to $\mathbf{m}$ is computed by

$$d^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m})^T \Sigma^{-1} (\mathbf{x} - \mathbf{m}) \tag{11}$$

and the new position $\mathbf{m}(t+1)$ is updated as

$$\mathbf{m}(t+1) = \frac{\sum_j \mathbf{x}_j \exp\{-\frac{1}{2}(\mathbf{x}_j - \mathbf{m}(t))^T \Sigma^{-1} (\mathbf{x}_j - \mathbf{m}(t))\}}{\sum_j exp\{-\frac{1}{2}(\mathbf{x}_j - \mathbf{m}(t))^T \Sigma^{-1} (\mathbf{x}_j - \mathbf{m}(t))\}} \tag{12}$$

where $\mathbf{x}_j$ is the j-th point within a certain distance to $\mathbf{m}(t)$. The new position is actually a weighted mean of the points with weights being their distribution probabilities.

We start from matching feature points like corners, junctions and spots. For motion images, since the difference is small, flows of feature points can be obtained by template matching. But in stereo and object recognition, in general correspondences have to be searched for in the whole image. Moreover, one has to take into possible fronto-parallel rotation into account. That is, template must be allowed to rotate in the image. Of course, for a group of points to share the same motion, the rotation angle should be the same. In this case, Template matching generally does not provide unique results. It means that more points will be generated in the motion parameter space that do not belong to any clusters. As far as they do not form clusters themselves, which is the case because of its randomness, they do not pose serious problems.

6 Matching Edge Points by Relaxation

Now suppose that we are to find correspondences for edge points. With known epipolar equations, we can find all possible pairs of edge point matches in the two images, each of which corresponds to an active node in SDS. The aim of matching is to leave only one active node for each edge point, and delete all other initial active nodes. As usual, we employ the smoothness constraint. We define an energy for each match,

$$E(x, y, d) = \sum_{i=-\Delta x}^{\Delta x} \sum_{j=-\Delta y}^{\Delta y} \frac{(\Delta d_{min}(x+j, y+j))^2}{i^2 + j^2} \tag{13}$$

where $\Delta d_{min}(x+j, y+j)$ is the possible minimal difference between $d(x, y)$ and other active nodes for $(x+j, y+j)$. The energy is updated for each iteration for each active node, and the node with the largest energy is deleted for each iteration till there is only one active node left for each edge point.

7 Experimental Results

We have done experiments for motion images with multiple rigid motions, uncalibrated model images and input image for object recognition. The first step

is to detect corners, junctions as feature points. One example of clusters in the motion parameter space is shown in Fig.1 for a pair of motion images with two motions, where the cluster centers are marked by circles. Fig.2-3 show a pair of motion images with two moving balls and the static background. The results of correspondence and segmentation for the two balls moving independently are shown in Fig.4-5. The background is also matched and segmented but not shown here due to space lack. Fig.6 shows a Mac computer, and Fig.7 shows the same computer in a room with cluttered background. The result of matching the model and input views with the correct epipolar equation is shown in Fig.8.

8 Conclusions

I have in this paper described a new unified approach to the problems of image matching in stereo, motion and object recognition by recovering the epipolar geometry underlying the images and changing the 2D search of correspondence to a 1D search along epipolar lines. All the three problems which have been treated in different frameworks can now be unified into a common framework. Segmentation is simultaneous. Algorithms for recovering epipolar equations, and using the recovered equations to match and segment images are described. Experimental results show that this approach does provide good results for motion correspondence and segmentation, and solves the unsolved problem of matching and segmenting uncalibrated views which are inevitably involved in 3D object recognition using model views instead of explicit 3D models.

References

1. J. Aloimonos. Perspective approximations. *Image and Vision Computing*, 8(3):177–192, 1990.
2. R. Deriche, Z. Zhang, Q. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In *Proc. Third European Conf. Comput. Vision*, pages 567–576, Vol.I, May 1994.
3. O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, MA, 1993.
4. W. Eric L. Grimson. *From Images to Surfaces.* The MIT Press, Cambridge, MA, 1981.
5. E.C. Hildreth. Computations underlying the measurement of visual motion. *Artif. Intell.*, 23:309–354, 1984.
6. B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artif. Intell.*, 20:199–228, 1981.
7. J.J. Koenderink and A.J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America A*, 8:337–385, 1991.
8. D. Marr. *Vision.* W.H. Freeman, San Francisco, CA, 1982.
9. J. L. Mundy and A. Zisserman, editors. *Geometric invariance in computer vision.* MIT Press, 1992.
10. E. Nishimura, G. Xu, and S. Tsuji. Motion segmentation and correspondence using epipolar constraint. In *Proc. 1st Asian Conf. Computer Vision*, pages 199–204, Osaka, Japan, 1993.

11. L.S. Shapiro, A. Zisserman, and M. Brady. Motion from point matches using affine epipolar geometry. In *Proc. Third European Conf. Comput. Vision*, 1994.
12. S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Trans. PAMI*, 13(10):992–1106, 1991.
13. P. Winston. *Artificial Intelligence, 3rd Edition.* Addison Wesley, Reading, MA, 1992.
14. G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach.* Kluwer Academic Press, 1996. to be published.
15. A. Yuille Y. Yang and J. Liu. Local, global, multilevel stereo matching. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.* IEEE, 1993.

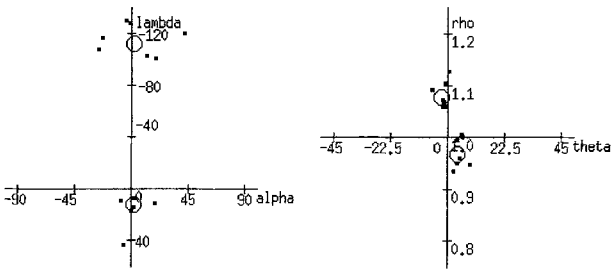


Fig. 1. Clustering in $\rho - \theta - \alpha - \lambda$ space

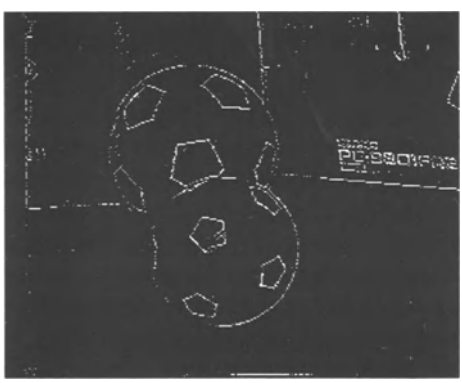


Fig. 2. Motion image 1

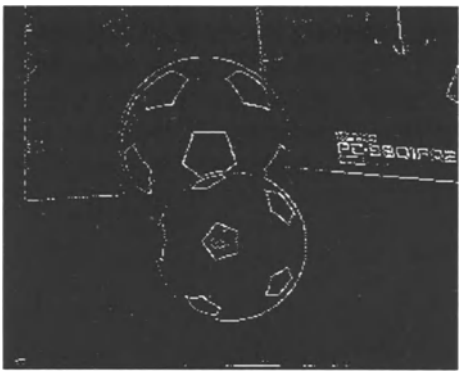


Fig. 3. Motion image 2

Fig. 4. Matched points for ball 1

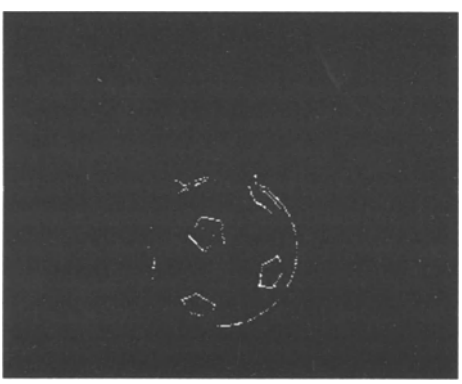


Fig. 5. Matched points for ball 2

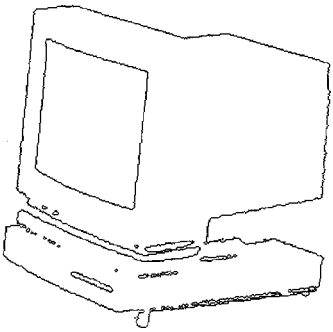


Fig. 6. Model image

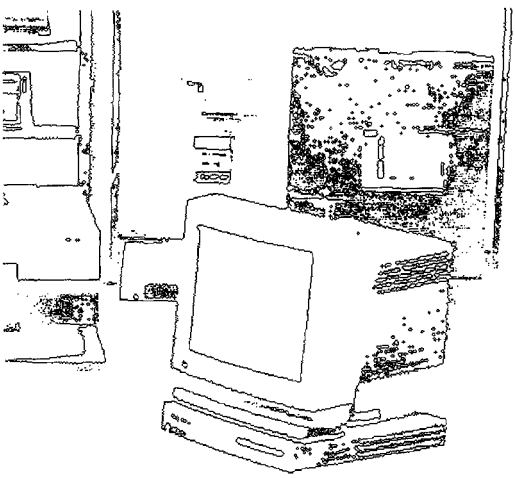


Fig. 7. Input image

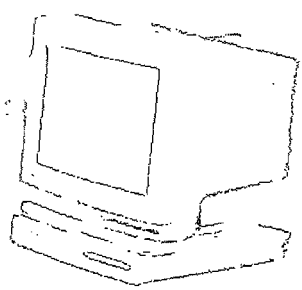


Fig. 8. Matched points in input image

Constructing a Visual Memory in Terms of Uniqueness

Jiang-Yu Zheng

Faculty of Computer Science and Systems Engineering
Kyushu Institute of Technology, Japan

Abstract

We propose a new scheme in representing and memorizing an image for pattern matching, scene understanding, and image database retrieval. A pattern in a visual space is evaluated in terms of uniqueness that reflects how stable and distinguishable the pattern is in its matching and recognition process. Features are accessed in the images in two fashions that are similar to saccade and smooth search in human vision, which makes the matching process to any particular feature be controllable. A view is organized in a structure linking unique patterns to less unique ones. Our uniqueness of a pattern is measured from both attributes of the pattern and its spatial relationship with others in its visual context. Establishing correspondence between two close views using this representation is not only efficient in accessing image primitives, but also reliable compared with conventional matching approaches that are based on restricted geometry constraints. It will be effective in active vision, memory based recognition, qualitative description and reasoning of scenes.

1 Introduction

This work deals with a problem of modeling a view for scene matching. A view can be either an image from camera or a graph with patterns. The same view is captured again with certain quantitative changes in attributes and geometry. What we attempt to do is to identify and locate the same patterns in the new visual space, assuming the contexts of the two views are not too different. This task (Fig. 1) is widely carried out in dealing with both short term visual memory such as stereo matching[1], tracking dynamic images[2], and long term visual memory such as scene recognition in an image database by a sample, target localization in aerial image or map, route scene identification for robot navigation[3,4], etc.

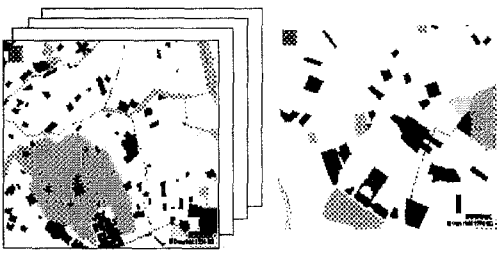
Most approaches of image accessing currently scan images line by line and extract attributes phase by phase. Global matching of images is based on attributes' distributions, while local matching uses some explicit geometry constraints to confine candidate search in a small area to select one with the highest similarity. The global approaches are stable but insensitive to local differences, while the local ones are hard to match two views with changeable properties and geometry, as well as repetitive patterns.

Human has excellent ability in matching views, even constraints such as epipolar line and motion smoothness are not available, and attributes for comparison such as color, size, and shape have slight changes. A new approach of matching which has capability beyond the numerical computation of similarities in a small constrained area is necessary. We will step towards this goal by first studying the visual memory that can cope with the view matching when viewing condition changes. The idea in this paper is to build a representation of scene to facilitate two new pattern searching approaches — saccade and oriented search in the images. To identify a view, some pattern is significant and some pattern is trivial. Some part of data is explicit and

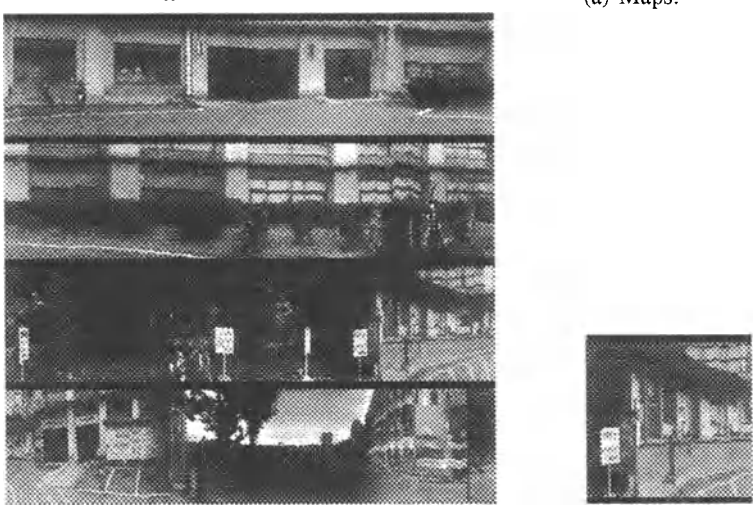
reliable while some part is ambiguous and cost in analysis. Even for an explicit pattern, if it has a repetitive distribution, one primitive is not as distinctive as its group. Besides high level image description that can guide image interpretation, we also need information conveyed from remarkable and stable parts in the same view. We hence represent patterns according to how unique they are in the visual context. The uniqueness is studied at low image level here. The topics to discuss are:

1. Measuring *uniqueness* of a feature from its attribute and spatial relation with others.
2. Evaluating significance of features in *attentive field* computed from the uniqueness.
3. Constructing a *map* to represent a view by selecting *landmarks* and their *accessing paths*.

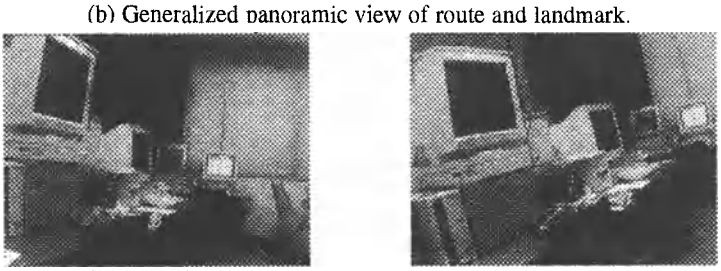
The consequential affects are as follows.



(a) Maps.



(b) Generalized panoramic view of route and landmark.



(c) Matching two close views can provide 3D and dynamic information.

Fig. 1 Matching views or patterns in the views in many visual tasks.

1. It constructs a structure so that sequential reasoning of patterns through saccade or oriented search becomes possible. Reliable patterns can therefore provide information to ambiguous patterns in a visual context.
2. The map allows us to access a pattern of interest stably with few steps. Visual processing can selectively work on some distinct pattern instead of the whole view, which is an important step towards qualitative vision that directly gives some brief answer.
3. The establishment of the memory is related to active vision[14,15]. An active camera needs to select a reference point (fixation point) and track it sturdily. If we can build an attentive field with uniqueness, active sensing strategy[9,10] is easy to be designed.

2. Uniqueness

What is uniqueness of a pattern? How it is able to be measured in a view? The first answer is how different the feature's attribute is from its neighbors. The second answer is how far away a similar feature appears, because a close one is easy be confused with the feature. Besides attributes such as color, size, shape, contrast, context information also plays an important role in the recognition. Intuitively speaking, a pattern that has a common color with others can still be distinctive, if it is spatially away from the similar ones.

Our uniqueness includes both local saliency and global saliency. The local saliency shows how strong a pattern contrasts with the background, and the global saliency shows how unique a pattern is in a large area. Segmentation (edge detection) is based on local saliency. Grouping is concerning with global saliency[11,12] as those similar patterns (with low global saliency) are grouped. We combine attribute and spatial measure for another objective — investigating how significant a pattern is against its visual context. In the computation of uniqueness, we compare attributes between a pattern and its neighbors, which yields edges, and compare its attribute with others at far away, which yields distinctive range of the pattern.

We limit the attribute in this paper to be intensity. A pattern is reached if the memorized gray level is found. Using more features such as size and shape will certainly improve the accuracy of image description. Pattern searching is carried out in two new approaches as follows (Fig. 2).

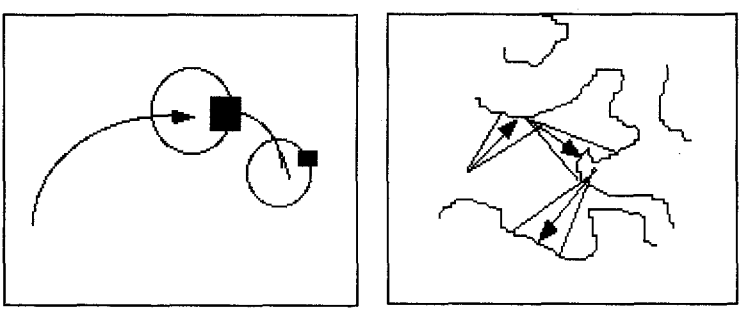


Fig. 2 Two ways of pattern search. (a) saccade, (b) oriented search.

Saccade: Given a relative vector, the focus point has a jump from the current position to a pattern of interest. If the expected attribute value is not found at the arrived point, it performs another search on circles expending from the arrived point until the value is detected.

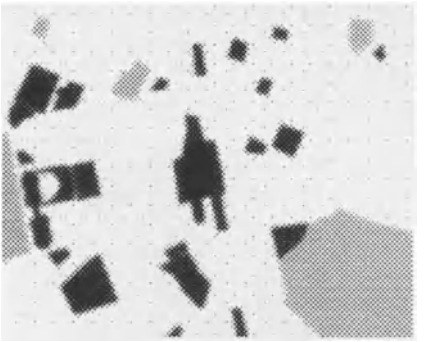
Oriented search: along a given direction, image is searched pixel by pixel from current position, until the expected value is found.

These two kinds of search use spatial relationship of patterns directly. They are close to what a human qualitatively uses in describing pattern locations. The move of focus point could be implemented either in 3D world by moving active head (robot) and eye (camera) [14,15,17] or in 2D images and the panoramic views[3].

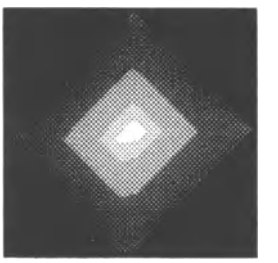
The further search in the saccade outward from the arrived position allows a pattern to have a small deviation in the second view from the predicted position. The saccade could be used to localize small and homogeneous patterns. The oriented search can locate remarkable boundaries of extending patterns.

3. Uniqueness of Homogeneous Patterns

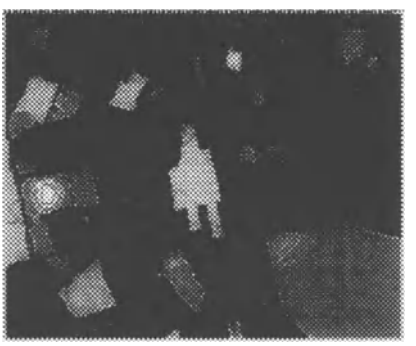
Suppose an image or a graph contains patterns that have homogeneous attribute values. Regions can therefore be detected and boundaries composed of edge points are closed around the regions. This kind of images can be maps, binary images, etc.



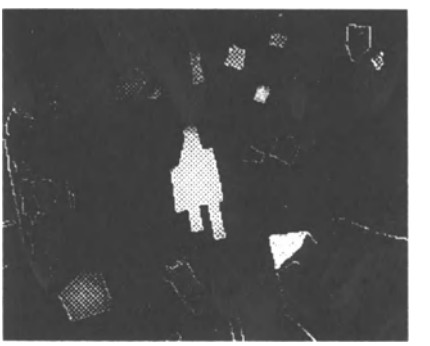
(a) A map with buildings and parks.



(b) Saliency distribution of a pattern. Small difference has dark value.



(c) Distinctive ranges of all patterns at level 0. The ranges are expanded until reach another pattern with the same intensity.



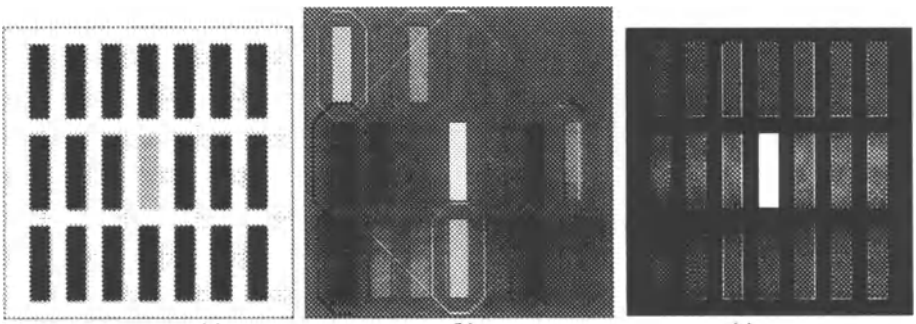
(d) Uniqueness of all patterns (bright patterns have high uniqueness).

Fig. 3 Measuring uniqueness of patterns in a graph.

For a segmented pattern, we compute the difference of attributes between the pattern and image points that have distance r from pattern's boundary. The minimum absolute value of the differences at the same distance is denoted as D(r). The boundary of each pattern is expanded pixel by pixel outward for the distribution D(r). By forcing such a saliency distribution in monotonous descending order as the distance increases, we obtain a *distinctive range function* F(r) of the pattern, as well as *distinctive range*

at any level of difference. The computation stops if the difference is less than a given threshold δ (a value under which attributes are unable to discriminate). The distinctive range means intuitively that within the distance r, there is no other pattern that has attribute closer to the focused one than δ, so that no pattern in the range r will be confused with the focused one if we search the attribute with a tolerance of δ.

The *uniqueness* of the pattern can be computed by accumulating values of the distinctive range function since we need the pattern to be salient in a large area. Figure 3 and 4 show two examples of uniqueness measure on homogeneous patterns.



(a) (b) (c)

Fig. 4 Repetitive patterns with an accent and measured uniqueness. (a) Original patterns. (b) Distinctive ranges of patterns. Patterns are assigned with different labels and their distinctive ranges are lined with the same colors. (c) Uniqueness of patterns.

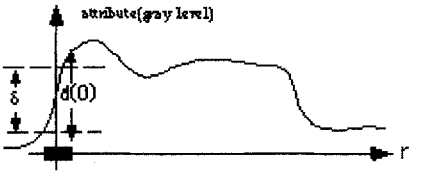
4. Uniqueness on Partial Boundary

Generally, a pattern is not insured to have a closed boundary in the image. The uniqueness in this case is hard to be defined on each region. We define uniqueness of patterns on partial boundaries for the oriented search. If a memorized intensity value near a boundary is found when searching from the other side of the boundary, the pattern is reached.

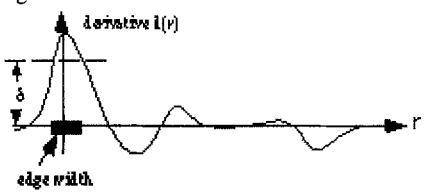
We first filter the image by a differential operator and detect edges with values greater than a given threshold δ. Values lower than the threshold are not considered since they do not satisfy local saliency criterion (Fig. 5.b).

For each edge point, its normal direction is determined and the difference between gray levels at two sides of the edge is registered as d(0). From the edge, several computations expand pixel by pixel along the normal directions as follows:

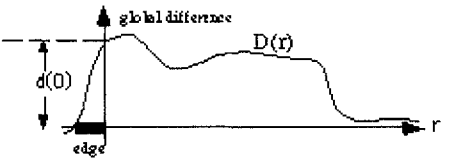
1. Calculating the difference of attribute between two consecutive points along the expending direction which is denoted as d(r) and r is the range of expansion.
2. Accumulating local difference d(r) to produce difference of attributes
$$D(r) = \int d(r)\, dr + d(0) \tag{1}$$
between the pattern and the expanded point. It gives global saliency of the pattern across the edge.
3. Forming a monotonous decreasing function F(r) from the absolute value D(r).
$$F(r) = \min |D(r)| \tag{2}$$
This ensures that any point closer to the pattern than r is different from the pattern more than F(r). We call F(r) *distinctive range function* of the pattern. The value of the function tends to zero if a similar pattern appears at that range.
4. Accumulating the distinctive range function to yield uniqueness of the pattern, which is the area under the distinctive range function expended so far.



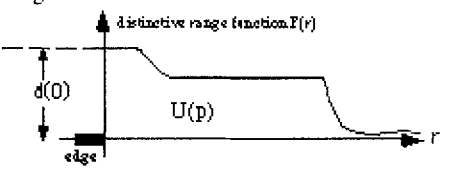
(a) attribute distribution along one of the normal directions of an edge when the edge is detected.



(b) Local saliency computed from derivative and an edge can be extracted if the local saliency is great than a threshold.



(c) global saliency computed from the difference between the point near edge and points far away at the other side of the edge.



(d) distinctive range function.

Fig. 5 Computing uniqueness of a pattern when it is accessed from the other side of edge along the normal direction.

The expansion for the above computation stops when the value of distinctive range function becomes lower than the threshold. The expansions in the opposite directions from the edge give uniquenesses of two patterns on both sides of the edge.

The computations are carried out along curves connected by edge points. In the real image, curves are tracked first. The expansions of a curve on its both sides are performed using chain code searching. The normal directions of edge points are figured out at the orthogonal direction of the chain along the expanding boundary. We do not restrict the oriented search of edge to be normal direction only but close to the orthogonal direction. Figure 6 tests the repetitive patterns and the uniquenesses on edges tell us the edges at the side patterns have the highest uniqueness. That means we can start matching of repetitive patterns from their side ones. Figure 7 gives the result showing the expansion and performed computations on patterns.

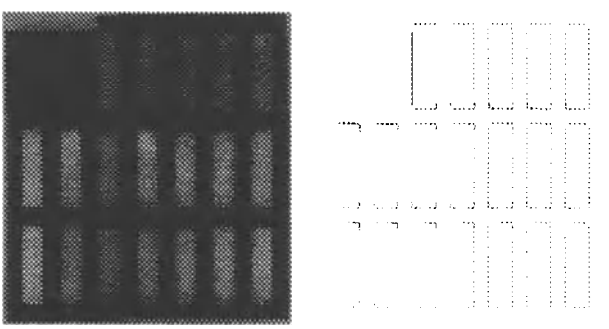
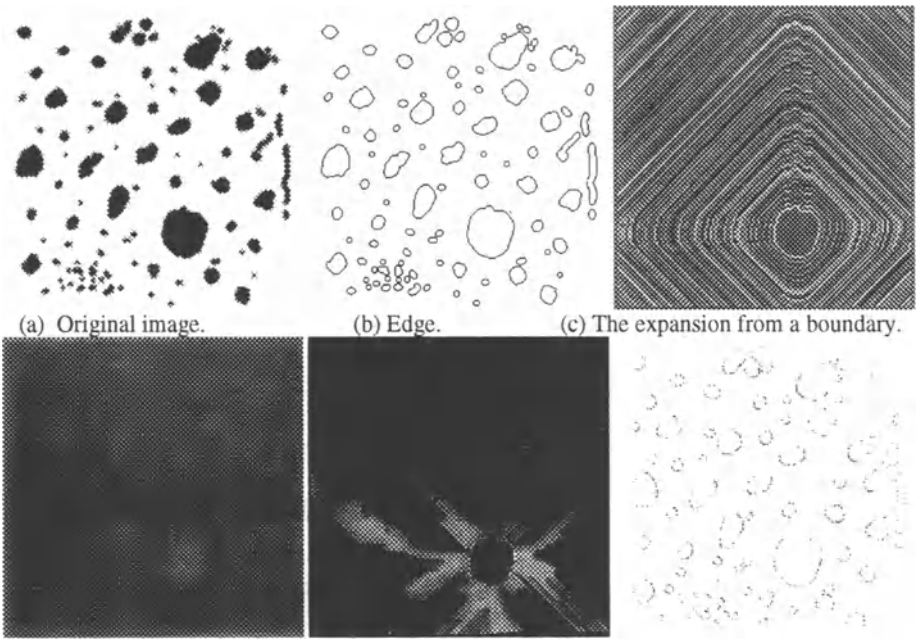


Fig. 6 Repetitive patterns and their uniqueness. (a) The expansion of global saliency computation. (b) Uniqueness of edges. The dark parts have high uniqueness.

5 Qualitative Visual Memory

5.1 Attentive Field

In the image recognition, the issue where to start matching[17] has not been studied much. Although there are many kinds of semantic information in the scene

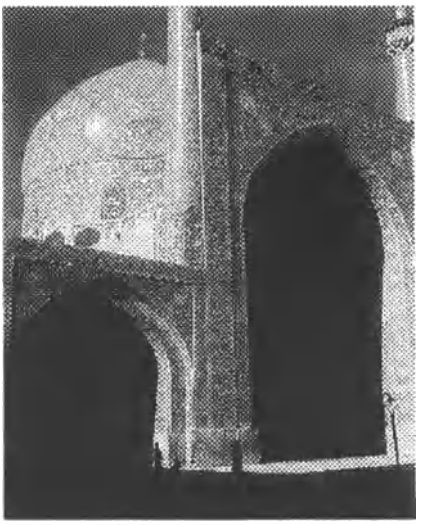


(d) Global difference in attribute between the pattern inside the boundary and points at all expanded positions. Other patterns with the dark values have differences from the focused one near zero (displayed in gray). The positive and negative differences are bright and dark respectively (most areas are negative). (e) Distinctive range function displayed in gray level. Zero value is dark. The expansion stops when reaching zero. (f) Uniqueness attached to all boundaries of patterns.

Fig. 7 Testing uniqueness computation from a boundary.

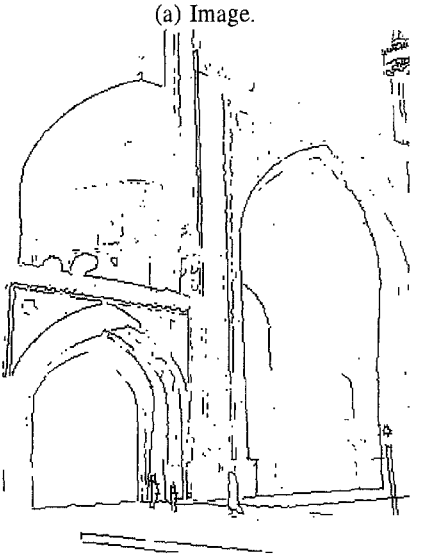
such as names and functions of patterns that guides the matching process, we believe uniqueness is a very good measure to evaluate how much a pattern attracts our eye. We generate an *attentive field* based on low level features. It is an image with value at each pixel as the uniqueness. For homogeneous patterns, the uniqueness is assigned to each region. In general images, non-edge points have certainly zero value because the uniqueness reflects local saliency first. Two uniqueness values at an edge point reflect how distinctive the two associated patterns are when they are accessed from the opposite normal directions of the edge respectively. Even for a strong edge, one of repetitive patterns for example, it may have a low value of uniqueness if an attribute on its sides is close to a similar one nearby. It is interest to notice that our uniqueness measure defined on edges assigns different values on a series of repetitive patterns. The pattern at the end of the series has a higher uniqueness than those in the middle, which implies we can start matching of such a sequence from the end patterns. It was impossible to be done if only looking at pattern's attributes without referring their spatial context and searching from some specific direction.

Figure 8 gives examples of attentive fields. In the first image, if we look at the right edge of the bright column at the center, we can say the column is unique than dark gates. The column has the distinctive range (approximately the width of the right gate) when it is searched from the right. It is wider then the distinctive range of right gate (approximately the width of the column). In the second image, we can notice that searching from the canal to the shore is more distinctive than locating the boundary of

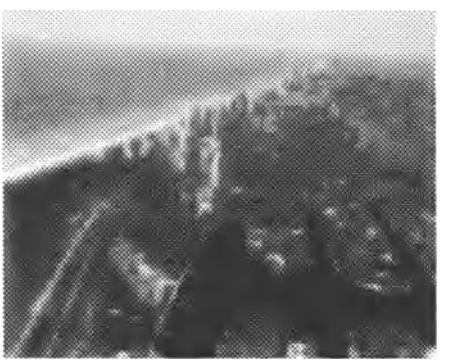


(a) Image.

(c) Attentive field measured at edges. High uniqueness is displayed in dark level. Two values are assigned to an edge point describing uniquenesses of patterns on two sides of it.



(b) Edges



(d) another image.

(e) Edge.

(f) attentive field.

Fig. 8 Attentive field measured by uniqueness of features.

the canal from the residential area. The attentive field of the images with homogeneous patterns has been given in Fig. 3.d and Fig. 4. We can first search those patterns with high uniqueness in making image correspondence and locating patterns of interest in the view.

5.2 Qualitative Visual Memory from Attentive Field

Now, we can discuss the issue of how to build a qualitative visual memory called *map* to access patterns in the image matching. Obviously, processing starting from patterns with high uniqueness is more reasonable and reliable than starting from an image corner. Unique patterns are less confusion than those ambiguous ones. Also, searching from a matched pattern to an unmatched one is more efficient than searching the whole image grid. Instead of determining the most similar candidate in a constrained region as a conventional approach does, which always fails when multiple similar candidates exist, we search patterns within their distinctive ranges. Patterns with high uniqueness are selected as *landmarks* in the map.

Let us first look at the qualitative memory of homogeneous patterns. For a particular pattern, we build a sequence of *paths* arriving at it from a starting point such as image center (image center or image corner is not important) via several distinctive patterns (landmarks). For 2D saccade, a jump of focus point has a region of deviation that is proportional in size to the jumped distance. A search of the desired pattern is followed expanding from the jumped destination. In the case of extending boundaries from general images, a direction of path from a unique boundary to a less unique one is computed. There is a deviation of angle in the oriented searching so that the frontier of the search becomes wider at far away.

The criteria of finding paths to a pattern from the most distinctive pattern are considered. First, the number of path segments should be small so that the landmark accessing has low cost. Second, the deviation of each segment should be enclosed in a tolerable range of the next landmark. This is a sufficient condition that insures the jump along such a path to reach the pattern of interest. If a pattern is unique, it can be reached through a long distance jump that saves the number of landmark accessing. Otherwise, a short jump with small deviation has to be connected from a unique pattern nearby. Figure 9 shows the net connecting patterns in the view given in Fig. 3. The result of the paths is not necessarily to be unique.

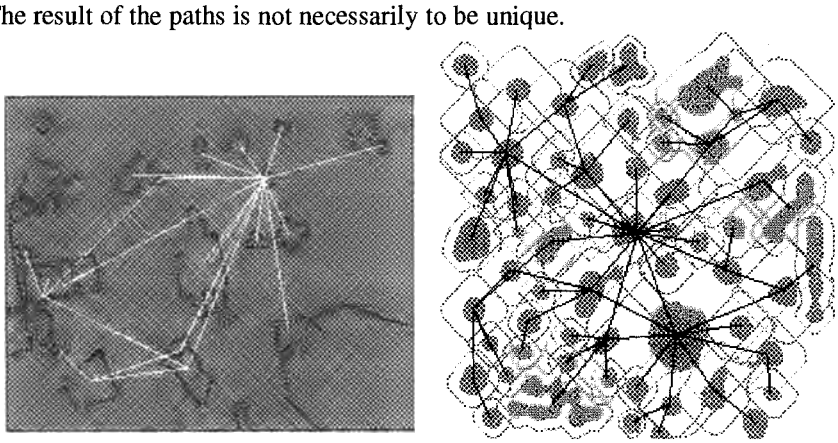


Fig. 9 Qualitative visual memory describing remarkable patterns connected by accessing paths. (a) paths to patterns via landmarks for the image in Fig. 3. (b) paths to all patterns in the image of Fig. 7, where pattern size is used as attribute.

6 Matching by Context

The matching strategy given here is called matching by context rather than conventional matching by constraints. Matching of pattern is determined not only from attributes but also from known patterns nearby in the space. View matching is a sequential process that finds correspondences of patterns from the most distinctive one to less distinctive ones.

Based on the visual memory from the first view, a predicted location or a searching direction to a new pattern from a landmark is given. The location and direction information are different from constraints such as epipolar line and motion smoothness. The real jump or search in the second view from an arrived pattern to a predicted position or direction may be inaccurate as the spatial relation in the first view is used. The real patterns in the second view may have change from the expected position due to changes in the view size, camera pose, and view point (in other words, scale changes, translation and rotation of view, and depth change). The jump arrives at a position near the pattern to match. Because the pattern to search is unique in the search area, there is no other pattern that will be mismatched.

The issue needs to explore is how much tolerance it has for each inaccurate jump or search. Besides using uniqueness in landmark selection, distinctive range can also guide the pattern matching. Let us examine the matching of two dimensional views using saccade. We set a threshold δ to accept a pattern P to match when a gray level within the $[I(P)-\delta, I(P)+\delta]$ is found ($I(P)$ is the gray level of the pattern P). The distinctive range at level δ is denoted as r. It ensures that no other patterns in the range will have difference from the P smaller than δ. If the jump arrives at a position that is inside the range of r/2 from the pattern, the circular search followed is guaranteed to reach the pattern P first rather than the boundary of the distinctive range of r. Therefore, the tolerable deviation is half of the distinctive range r at the given level of discrimination.

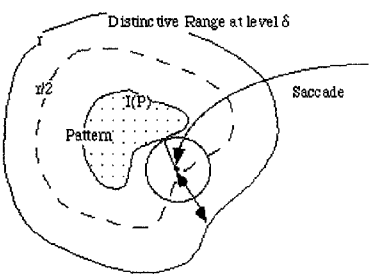


Fig. 10 If the saccade can reach the range of r/2 inside the distinctive range, the pattern to match will be found stably. A unique pattern has a large distinctive range to insure the matching to be correct.

7. Discussion and Conclusion

The uniqueness can be defined on each level along the category of abstraction. We can study this paradigm from low level data such as distribution, regions, edges to high level primitive such as structure, object, and function of object. The idea can also be used to more symbolic primitives such as line, curve, surface, volume, etc.

It is true that the image modeling using uniqueness needs more computation than edge detection. However, the computation in image retrieval is saved when we selectively search landmarks. Even in the image modeling, the following situations, fortunately, save much computation. The expansion for the distinctive range stops at a near range if a pattern is not unique. Only for those unique patterns, the computation extends to far away. Second, in many situations, we are only interested

in several particular patterns. We do not have to build a memory containing every detail in the view.

We described a new scheme to represent visual data. Based on the idea of uniqueness, a qualitative memory to guide the matching of pattern can be constructed. The context based matching is robust to small changes both in attributes and geometric location for distinctive patterns. We measure uniqueness of a pattern from its local saliency and global saliency, which obtains spatial relation of pattern with others. The obtained attentive field is useful for selection of focus point in active vision paradigm. We are now matching real images using this representation. In addition to the homogeneous regions and edges, we will define uniqueness on more features such as shape, size, and semantic information and give some evaluation of performance in matching using this new paradigm.

References

[1] Y. Ohta and T. Kanade, "Stereo by two-level dynamic programming" Proc. IJCAI-85, Vol.2 pp.1120-1126, 1985.

[2] M. Yachida, M. Asada, and S. Tsuji, "Automatic analysis of moving images", PAMI, Vol.3, No.1, pp 12-19, 1981.

[3] J.Y. Zheng and S. Tsuji, "Panoramic representation for route recognition by a mobile robot", IJCV, Vol.9, No.1, pp.55-76, 1992.

[4] J.Y. Zheng, M. Barth and S. Tsuji, "Qualitative route scene description using autonomous landmark detection", Proc. 3rd-ICCV, pp. 558-562, 1990.

[5] P. J. Burt, "Smart sensing within a pyramic vision machine", IEEE Proceedings on Computer Vision, pp.1006-1015, 1988.

[6] A. Witkin, "Scale-space filtering", in IJCAI, Karlsruhe, pp.1019-1021. 1983.

[7] P. Perona and J. Maliik, "Scale-space and edge detection using anisotropic diffusion", IEEE Trans. on PAMI, Vol. 12, No.7, July 1990.

[8] T. Lindeberg, and J. Eklundh, "Scale detection and region extraction from a scale-space primal sketch", Proc. 3rd-ICCV, pp.416-426, 1990.

[9] D.H. Ballard, "Reference frame for animate vision", Proc. IJCAI-89, pp.1635-1641 (1989).

[10] R. Bajcsy, "Active perception", Proc. IEEE special issue on Computer Vision, August 1988, pp.996-1005.

[11] N. Ahuja and M. Tuceryan, "Extraction of early perceptual structure in dot patterns: integrating region, boundary, and component Gestalt", CVGIP 48, pp.304-356, 1989.

[12] A. Sha'ashua and S. Ullman, "Structural saliency: The detection of globally salient structures using a locally connected network", Proc. 2nd-ICCV, pp.321-327, 1988.

[13] G. Cohen, Memory in the Real World, Lawrence Erlbaum Associates, Hove. 1989.

[14] J. Y. Zheng, Q. Chen and S. Tsuji, "Active camera guided manipulation", 1991 IEEE Int. Conf. Robotics and Automation, Vol. 1, pp. 632-638, 1991.

[15] J. Y. Zheng, M. Barth, and S. Tsuji, "Landmark detection for route recognitioni by a mobile robot", 1991 IEEE Int. Conf. Robotics and Automation, Vol. 3, pp.2004-2009, 1991.

[16] B. Dubuc, and S. W. Zucker, "Indexing visual representations through the complexity map", 5th ICCV, pp.142-149, 1995.

[17] M. Jagersand "Saliency maps and attention selection in scale coordinates: an information theoretic approach", 5th ICCV, pp.195-202, 1995.

Statistics and Image Analysis

Statistical Models & Algorithms for Tomography

Malcolm Hudson *

Department of Statistics,
Macquarie University, NSW 2109
Australia
mhudson @ mq.edu.au

Abstract. In this paper we describe various tomographic methods used in medical imaging of patients, describing the forms of the data collected and statistical algorithms for reconstructing the image. Our goal is an expository paper which highlights some statistical questions and contributions in medical imaging.

1 Medical Imaging

The general objective of tomography methods to map structure *within* the body from observations *outside*. Modern medical computerised methods include:

1. Gamma camera imaging
2. Computer (transmission) tomography (CT)
3. Emission tomography
   (a) single photon emitter (SPECT)
   (b) positron (two photon) emitter (PET)
4. Magnetic resonance imaging (MRI)

Transmission and emission tomography have different goals and specific data acquisition techniques using varied instrumentation but are very similar in the mathematical modelling of their data. This common model means similar reconstruction techniques may be applied in either application.

To be specific in describing objectives and data acquisition we shall briefly consider an illustrative setting from single photon emission tomography. The context will be a radionucleide introduced into the patient's bloodstream and thereafter distributed throughout an imaging region within the body, producing radioactive decay events, emitting single photons.

1.1 Single Photon Emitters

SPECT recordings have several elements, introduced below.

* Supported in part by Australian Research Council grants.

Camera and collimator: The camera records photons impacting on the camera crystal originally generated as *events* in the body region (e.g. nuclear disintegrations of a tracer taken up in the body). Photons are recorded by the camera only if they pass through the bore holes of a lead collimator designed to accept only photons originating within a specific limited region within the body (e.g. from voxels directly aligned with the bore hole; with parallel bore collimators we shall refer to voxels *opposite* a camera pixel).

Projection: These recordings on the camera are *binned*, producing the spatial distribution of recordings as pixel counts. These counts form a 2d histogram and may be viewed as an image, a map of spatial distribution of recordings on the camera. The mathematical operation of computation of expected projection data based on a given source distribution is also termed *projection*.

Angular sampling: In many systems a gamma camera acquires data, then rotates around the patient to a new position. Camera rotation results in acquisition of a sequence of such images (*individual projections*).

Images display the *distribution* and *intensity*, of photon activity. If collimation is fully effective the activity in any slice (plane) normal to the axis of camera rotation is recorded in the same plane. We may then restrict attention to a sequence of 1d histograms, pasted together as rows of a 2d *sinogram*.

The goal of reconstruction is to infer from the camera acquired data the distribution within the body of the source activity.

2 Statistical Contributions

2.1 Statistical issues

A number of statistical issues arise within tomography applications. A recent review is McColl et al [7]. In brief, some areas of interest are:

Statistical reconstruction methods: Reconstructions are not only used for clinical studies, providing a basis for individual patient diagnosis (e.g. tumors, lesions, blood flow studies), but provide the raw data for research studies described below.

Image fusion: MRI can provide very accurate maps of *anatomy* while emission tomography provides much lower resolution maps of *function*. Linking function to the anatomy is of interest. This requires image registration in order to align the co-ordinate systems of the two maps.

Research study design and analysis: Common statistical designs present unusual analysis difficulties with imaged data. For instance, images for different subjects may have to be transformed to a 'prototype' anatomy before effective comparisons can be made. Two examples showing simple designs are:

1. comparing imaged response in patient groups defined by disease status (cross-sectional);
2. Activation studies - continued imaging of a subject given a series of tasks (repeated measures).

Dynamics: mapping body metabolism through time varying uptake of a tracer introduced into the bloodstream in a controlled manner. Time sequences of images result, with the aim of reconstruction being to provide maps of parameters of models specifying metabolic activity in time.

2.2 Model

A very valuable feature of the tomographic problems described is that they can be subsumed as special cases of a general stochastic model (Snyder [10]) for which statistical reconstruction proceedures exist. The key features are:

- Emitter activity is well modelled as a non-homogeneous Poisson point process in the body region $\mathcal{X}$ with intensity proportional to unknown activity map ("source image")
- Camera recorded activity is modelled as a derived Poisson process on the camera domain. Poisson counts on one, or multiple, *projections* result. Each camera pixel cumulates source activity within the 'ray' opposite. Recorded activity then provides indirect information about the source activity.
- The link between source and camera images is defined by a known transition probability matrix known as the projection matrix.

Projection: Let $X \in \mathcal{X}$ be the voxel of the source event, $Y \in \mathcal{Y}$ be the pixel of projection, for a single emitted photon.

Transition probabilities or point spread function (psf) specify the conditional probs

$$a(t|s) = Pr\{Y = t|X = s\}$$

If $f(s)$ is the (unknown) distribution of activity, then the distribution of recorded activity is

$$g(t) = \int_{s \in \mathcal{X}} a(t|s) f(s) ds$$

or (in discretised form)

$$g_t = \sum_s a_{ts} f_s$$

Our objective: solve for f, knowing a and g

Note:

- Conditional probs $a(t|s)$ are determined by camera resolution, *geometry*, and the physics of photon transport through the body.
- There is great generality in this specification, whether in CT, SPECT or PET, for precise modelling of physical effects (attenuation, scattering) including ability to incorporate information from other modes.
- An issue for effective reconstruction is the high dimensionality - matrix $A = (a_{ts})$ is often 128^3x128^3, and of higher dimensionality still when applied to dynamics. Nevertheless, in tomography, simplifying assumptions specifying $a(t|s)$ as sparse (e.g. $a(t|s) = 0$ unless t 'opposite' s) are effective.

2.3 EM Algorithm

The distribution $g(t)$ corresponds to observable projection data. The Shepp-Vardi [9] application of Expectation Maximization (EM) in PET is easily translated to SPECT (Lange and Carson [5]) where it provides a reconstruction procedure that *re-allocates* $g(t)$ to pixels 'opposite' detectors t. This reallocation is termed *Back-projection*.

Back projection:

$$\hat{f}(s) = \int w(s|t)g(t)dt$$

Here the ideal is Bayes-reallocation: substitute

$$w(s|t) = \frac{a(t|s)f(s)}{\int a(t|u)f(u)du}$$

in back-projection. Unfortunately weights depend on unknown f.

The EM algorithm adopts instead iterative improvement: starting with $f^0 \equiv 1$, backproject in iteration $n+1$ by

$$f^{n+1}(s) = \int w^n(s|t)g(t)dt$$

Weights chosen mimic Bayes-rule (with current estimate):

$$w^n(s|t) = \frac{f^n(s)a(t|s)}{g^n(t)}$$

using fitted projections

$$g^n(t) = \int a(t|u)f^n(u)du$$

The result is the Shepp-Vardi EM algorithm:

$$f^{n+1}(s) = f^n(s) \int \frac{g(t)}{g^n(t)} a(t|s)dt$$

The points to note are that EM places photon counts on the camera back into the body using Bayes rule, and works from an initial guess about the source distibution, which is successively improved in iterations.

Computing EM: Computations require repetition of two steps.

1. *Project* the current source estimate to produce *fitted* projection data.
2. *Backproject* the ratio between *observed* and *fitted* projections, $g(t)$ and $g^n(t)$, to determine multiplicative corrections.

Properties: The EM algorithm converges from any initial $f^0 \geq 0$. This is a consequence of each step decreasing KL information distance

$$\int g(t) \log \frac{g(t)}{g^n(t)}$$

Applications: The Shepp-Vardi EM algorithm may also be modified to accomplish reconstruction adopting prior information in a Bayesian formulation (e.g. Green [3]) or other regularization methods. Another modification is the ordered subsets (OSEM) algorithm [4] whose aim is to accelerate EM in its original and Bayesian forms. Yet another form of EM is suitable with staged levels of incomplete data, a form particularly useful in dynamic models (Carson and Lange, [1]).

3 Comparative Studies

Comparisons of algorithms, e.g. EM with OSEM, can be conducted using data generated in simulation models, using data from camera projections of known source activity ('phantoms'), or using data from camera projections in a clinical context.

In simulations and phantom studies exact projections $g(t)$ of a *known* source $f(t)$ are computed for repeated projection angles. Simulations generate data with additional (Poisson) variabilty simulated from $g(t)$.

We measure discrepancies between: (i) data and fitted; (ii) source and reconstruction.

Measures of quality of fit to data (chi-square), and closeness to known source (mean square error) are available as:

$$CSQ = 2\{\sum y_i \log(\frac{y_i}{\hat{y}_i}) - \sum(y_i - \hat{y}_i)\}$$

$$MSE = \sum(\hat{x}_i - x_i)^2$$

4 EM applied in dynamics

We adopt the mathematical model of emission tomography of Shepp and Vardi [9]. This defines a time series of Poisson variables with expected counts satisfying linear models in parameters (the activity distribution of interest). Compartmental models incorporating a known input function $x_0(t)$ driving the dynamics specify a mixture model for emitted activity:

$$\lambda_{jt} = \sum_{k=1}^{K} \alpha_{jk}\ x_k(t; \sigma_k)$$

where

$$x_k(t) = x_0(t) * (1/\sigma_k) exp(-t/\sigma_k)$$

and $\alpha \geq 0, \sigma > 0$, for the expected number of events in pixel j, time interval t, and k indexes the modes of the model. Here $*$ denotes convolution. The model specifies spatially inhomogeneous (but piecewise uniform) activity in subregion (alternatively pixel) j of the image. The time activity curve (TAC) for each region is thereby assumed to be of semi-parametric form as a mixture of distributions (each a convolution of the input functions with an exponential distribution). Compartmental models (solutions of linear differential equations) for physiology are a special case. Alternatively, smoothing spline bases could provide the modal distributions x_k.

Rather than estimate a particular compartmental model parameters $\theta = (\{\alpha_k\}, \{\sigma_k\})$ Cunningham and Jones [2] propose a semi-parametric spectral decomposition useful in identifying among a broad selection of compartmental models. In this approach the model above is used, but with a number of significant modifications:

- the parameter vector α_j is assumed to be strictly positive;
- the distributions $x_k(t)$ are assumed completely known;
- whereas compartmental models usually employ small K, the spectral decomposition employs a large number, K, of possible modes. These correspond to a predetermined, fixed selection of possible choices for σ_k.

Note

1. The methodology is nonparametric. No specific compartmental model is assumed. Cunningham and Jones provide a number of illustrations of the interpretability of such models.
2. Sparsity constraints are required for identifying parameters. There will be a limited number of modes present in any one TAC. So we expect many components of α_j will be zero.
3. Without such sparse representation, interpretation in terms of compartmental models is complex.
4. The method can also be applied to TACs of indirect observations (projection data) equally well to determine significant modes.

Cunningham and Jones modeled *directly observable* recorded counts in a region of interest using *constrained* weighted least squares reconstruction for *positivity constrained* mixing parameters. They demonstrated empirically that this procedure produced sparse representation (mostly zero coefficients) in examples they considered.

O'Sullivan [8] provides procedures for fitting mixture models to reconstructions obtained by Filtered Back Projection.

In our current project we are providing EM algorithms for *indirectly observed* data, applicable with projection counts in tomography. EM algorithms are a natural choice since the model specifies a linear system for which Shepp and Vardi's method is, in principle, applicable. Furthermore EM algorithm iterations enforce positivity constraints, improving estimation quality.

5 Conclusion

Image data blurring and reconstruction are classical problems but new statistical reconstruction methods are being adopted because of the flexibility of their application, their failsafe behaviour and their ability to model physics accurately.

The statistical parameter of interest is of infinite dimension, a function of 2, 3 or 4 variables defining the distribution of source activity in space and time. Statistical methods can make use of Bayesian priors by specifying covariance structures for random effects (such as detector efficiencies) and local smoothness conditions (e.g. Gibbs priors applied within anatomically defined regions) to provide appropriate regularization. Computational issues become very important at this level of complexity.

The EM algorithm is very generally applicable in tomography, and, more widely, in linear inverse problems [6]. Subdivision of data in balanced blocks (OSEM) accelerates its convergence. The application of such approaches to new contexts, including improved physical modelling of scatter in SPECT and accelerated dynamic modelling in tomography is an area of current interest that will enhance the practicability of the EM algorithm.

References

1. Richard E. Carson and Kenneth Lange. The EM parametric image reconstruction algorithm. *J. American Statist. Assoc.*, 80:20–22, 1985.
2. Vincent J. Cunningham and Terry Jones. Spectral analysis of dynamic PET studies. *J. Cerebral Blood Flow and Metabolism*, 13:15–23, 1993.
3. P.J. Green. Bayesian reconstruction from emission tomography data using a modified EM algorithm. *IEEE Trans. Med. Imaging*, MI-9:84–93, 1990.
4. H. Malcolm Hudson and Richard S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans. Med. Imaging*, MI-13(4):601–609, 1994.
5. K. Lange and R. Carson. EM reconstruction algorithms for emission and transmission tomography. *J. Comp. Assisted Tomography*, 8:306–316, 1984.
6. Jun Ma. Maximum penalized likelihood solutions to linear inverse problems with generalized error distributions. Thesis, submitted for Ph.D. degree, Macquarie University, 1995.
7. J. H. McColl, A. P. Holmes, and I. Ford. Statistical methods in neuroimaging with particular application to emission tomography. *Statistical Methods in Medical Research*, 3(1):63–86, 1994.
8. F. O'Sullivan. Imaging radiotracer model parameters in PET: a mixture analysis approach. *IEEE Trans. Med. Imaging*, 12:399–412, 1993.
9. L.A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, MI-2:113–122, 1982.
10. D.L. Snyder and M.I. Miller. The use of sieves to stabilize images produced with the EM algorithm for emission tomography. *IEEE Trans. Nuc. Scie.*, NS-32:3864–3872, 1985.

Shape Statistics and Image Analysis

K.V. MARDIA
University of Leeds,
Department of Statistics,
Leeds LS2 9JT, U.K.
k.v.mardia@leeds.ac.uk

1 Introduction

Objects are everywhere - natural and man-made. The study of their shape can use control points if they exist but this is not always the case. The statistical methodology required for such work is inherently non-Euclidean, i.e. non-linear leading to the new subject of shape analysis. There have been various recent advancements in the fields of shape analysis and image analysis (see, for example, [29], [24], [30]) and we give a brief review of the work with emphasis on high level Bayesian image analysis.

2 High-level Bayesian Image Analysis

In the last fifteen years statistical approaches to image analysis using the Bayesian paradigm have proved to be very successful. Initially, the methodology was primarily developed for low level image analysis but is increasingly used for high level tasks.

Using the Bayesian paradigm one requires a prior model which represents our initial knowledge about the objects in a particular scene and a likelihood model which is the joint probability distribution of the image, dependent on the objects in the scene. By simply using Bayes Theorem one obtains the posterior distribution of the objects in the scene, which can be used for inference, e.g. segmentation and classification.

In low-level image analysis the prior could be say an Ising model, specifying that nearby pixels will tend to have similar grey levels i.e. the scene is composed mainly of large homogeneous objects. See for example [3], [11] and [15] for details of the methodology. Indeed, this approach is often referred to as a 'context' based approach in the remote sensing literature where the use of 'context' means that neighbouring **pixel** information has been used (eg. [12], [9], [31]). As we use 'context' to mean neighbouring **object** information our approach will concentrate on high level image analysis tasks such as object recognition. However, neighbouring pixel information will still be incorporated in the image model.

An appropriate method for high-level Bayesian image analysis is the use of deformable templates and our description follows the common theme of [28]. Consider our prior knowledge about the objects under study to be represented by a parameterised ideal prototype or template S_0. Note that S_0 could be

a template of a single object or of many objects in a scene. A probability distribution is assigned to the parameters with density (or probability function) $\pi(S)$, which models the allowed variations S of S_0. Hence, S is a random vector representing all possible templates with associated density $\pi(S)$. It is through the prior model that we can express the contextual knowledge - for example in face recognition the location of the mouth must be approximately half way between the eyes but not on the same level.

In addition to the prior model the image model is required. Let the observed image F be the matrix of grey-levels $f(t_i)$ where $t_i \in \mathbb{R}^2, i = 1, ..., N^2$ are the $N \times N$ pixel locations. This image model or likelihood is the joint probability density function of the grey levels given the parameterised templates S, written as $L(F \mid S)$. It expresses the dependence of the observed image on the deformed template. It is often convenient to generate an intermediate synthetic image $G = \{g(t_i), i = 1, ..., N^2\}$ which specifies how the parameterisation of S determines an image. The specification could be the mean grey value in a region or texturing within regions, and may need to take into account aspect and occlusion in projections of three dimensional objects. The intermediate image may be generated deterministically written as $G(S)$ or probabilistically according to the density $\pi(G \mid S)$. The observed image differs from G due to, for example, noise or blurring and so does not depend directly on S, except through G, so that $L(F \mid G) = L(F \mid G, S)$. In many applications G is the reconstructed 'true' image and S is the interpretation.

By Bayes Theorem, the posterior density $\pi(S, G \mid F)$ of the deformed template S and generated image G, given the observed image F is proportional to

$$L(F \mid G)\pi(G \mid S)\pi(S) \tag{1}$$

if the intermediate image is generated probabilistically, and

$$L(F \mid G(S))\pi(S) \tag{2}$$

if deterministic. Note that sometimes the construction of an intermediate image is not necessary and we have

$$\pi(S \mid F) \propto L(F \mid S)\pi(S). \tag{3}$$

In all these cases the solution to maximising the expression with respect to S and G is the maximum a posteriori (MAP) estimate of the true scene. The MAP is found either by a global search (which is often impractical due to the large number of parameters) or by techniques such as simulated annealing [11].

3 Prior Models

The key to successful inclusion of context in Bayesian image analysis is through successful specification of the prior distribution. A variety of approaches have been proposed, including methods based on outlines, landmarks, geometric parameters and Gibbs distributions. The prior can be specified either through a model with known parameters or with parameters estimated from training data.

3.1 Outline Based Approach

Consider a single object in $\mathbb{R}^2$ with k landmarks. Grenander and co-workers have constructed a general framework for image understanding using deformable templates. Most frequently they specify a series of points around the outline of the object and these are connected by straight line segments. Variability in the template is introduced by premultiplying the line segments by random scale-rotation matrices T_i,

$$T_i = \begin{bmatrix} u_i & v_i \\ -v_i & u_i \end{bmatrix} \quad , \quad i = 1, \ldots, k.$$

[16] choose first order circular autoregressive (CAR) processes independently for u_i and v_i. The application considered was locating a hand in a noisy image, with landmarks at finger tips and points in between (pseudo-landmarks). A few of the many applications considered by Grenander's group include identification of leaves in noisy images with landmarks at points of high curvature by [22] and models for chairs and human stomaches [17]. [27] review certain aspects of this work and give the corresponding distribution of the CAR model on the landmarks. [21] give details of the underlying conditional cyclic Markov random field.

Sometimes there may be no identifiable landmarks on an object and it may be convenient to take a number of equally spaced points around the outline of the object. [18] use this method for locating mitochondria on micrographs. In this case they assume a block circulant Toeplitz covariance matrix for $(u_1, v_1, ..., u_k, v_k)^T$ to construct a Gaussian model. A feature of this application was that the number of mitochondria in the image was unknown and so a random number of objects was included in the parameterisation. [13], [2] and [33] also consider Bayesian models with unknown numbers of objects. Another use of deformable templates was by [34] who used a random number of line segments of fixed lengths, joined at angles θ_r to model the arms of galaxies. The number of segments was modelled to be binomial and the angles were conditionally normal; a conditional von Mises is an alternative for the angles.

3.2 Landmarks: Principal Components Analysis

Consider again a single object in $\mathbb{R}^2$ with k landmarks. One approach to specifying the template is to work with the landmarks directly. Denote the coordinates of the landmarks in $\mathbb{R}^2$ as X a $2k \times 1$ vector. The configuration can be parameterised as an overall location (x_c, y_c) (eg. centroid), a scale d (eg. length between landmarks 1 and 2) and shape coordinates U (eg. the coordinates of the remaining points after translating, rotating and rescaling so that landmark 1 $\rightarrow$ (0,0) and landmark 2 $\rightarrow$ (1,0) - Bookstein's [5] coordinates). Under the multinormal model,

$$X \sim N(\mu, \Sigma), \tag{4}$$

[10] have given the shape distribution of U which could be used as a prior together with more vague priors for location, scale and rotation. However, in general there are many parameters and this approach is complicated. A simple form for Σ is required and a principal component (PC) approach has proved to be very effective.

[7], [8] and [19] consider the point distribution model which is a principal component model for shape. We formulate their principal component model for configurations as

$$X = \mu + \sum_{i=1}^{p} y_i \gamma_i + \epsilon \tag{5}$$

where $y_i \sim N(0, \lambda_i)$, $\epsilon \sim N(0, \sigma^2)$, independently and

$$\mu^T \gamma_i = 0, \gamma_i^T \gamma_i = 1, \gamma_i^T \gamma_j = 0, \quad (i \neq j),$$

and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$. Here $p \leq k$ and p is preferably taken to be quite small.

Note that, this approach allows us to give a model for flexible varying shapes, with often interpretable principal components. The interpretation of each component can be seen by varying y_i in equation (5) while fixing the other $y_j, j \neq i$, equal to zero. In practice the population parameters must be estimated from a random sample. Let $\hat{\mu}$ and $\hat{\gamma}_i$ be the estimates of μ and γ_i. Then $X = \hat{\mu} + y_i \hat{\gamma}_i$ for different y_i (but fixed i) captures different aspects of shape variation, eg. relative length, relative width, bending, movement of a subset of landmarks relative to the rest and so on. The principal component model is a special case of the general normal model (4) where the covariance matrix is

$$\Sigma = \sigma^2 I_{mk} + \sum_{i=1}^{p} \lambda_i \gamma_i \gamma_i^T.$$

As stated above marginal shape distributions are available in [10]. Therefore, inference for large samples could proceed using likelihood based methods.

However, this is complicated in general and so Procrustes methods were used for the situation where the variability in the dataset is small. Given l independent configurations $x_1, ..., x_l$, estimates of the above parameters can be obtained by first registering the shapes by Generalised Procrustes Analysis (GPA) and then performing a principal component analysis. In GPA each figure is translated, rescaled and rotated relative to each other so as to minimise the sum of squared Euclidean norms between figures. [8] use a type of weighted Procrustes analysis to obtain registered figures $x_1^P, ..., x_l^P$. The estimate of μ is taken to be the Procrustes mean $\hat{\mu} = \sum x_i^P / l$ and $\hat{\gamma}_i, i = 1, .., min(l, k)$ are the eigenvectors of

$$\frac{1}{l} \sum_{i=1}^{l} (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$

with corresponding decreasing eigenvalues $\hat{\lambda}_i$. Given a training set of data this approach allows us to give a model for flexible varying shapes (for example, see [7], [30]).

3.3 Geometric Parameter Approach

An alternative approach is to provide a geometric template consisting of parametric components, eg. line segments, circles, ellipses, arcs etc. For example, [34] use a circle of random radius for the central disc of galaxies. Also [32] use simple geometric shapes for facial features, pioneered by [38] and [39]. [2] use

circular templates to locate pellets in an image, where the number of pellets is unknown.

Models are specified for the geometrical parameters and the hierarchical approach of [32] seems particularly appropriate for context based vision. Often templates are defined by both global and local parameters. The global parameters are on a more coarse scale and the local parameters are more detailed measurements. The idea of a hierarchical model for templates is to specify the marginal distribution of the global parameters and a conditional distribution for the local parameters given the global values. This division of parameters can be continued to give a complete description of the local dependence structure between variables. Hence, conditionally each parameter depends only on variables in the levels immediately above and below it.

For example consider the mouth template of [32] after [39]. They use marginal normal distributions for (x, y) (location), θ (rotation), b (half the width) and conditional normal distributions for $a \mid b$ (height given halfwidth), $c \mid a$ (depth given height) and $d \mid a$ (curvature of parabola given height). Here x, y, θ and b are global, and the rest are local.

In more complicated image scenes where several templates are required the organisation of the templates can be considered at a higher level of hierarchy. For example, there may be nesting relationships between the templates, which are subject to constraints. For example, with human face templates there are global constraints such as the eyes, mouth and nose which must be strictly contained within the head boundary.

3.4 Gibbs Distributions

The Gibbs distribution has a probability density function of the form

$$\pi(s) = \frac{1}{Z} \exp(-\frac{1}{2} E_{int}(s, s_0)) \tag{6}$$

and has been used widely in image analysis. Of course any probability function can be expressed in this form but simple expressions for $E_{int}(s, s_0)$ will be of particular interest. The term $E_{int}(s, s_0)$ is called the internal energy function and Z is the integrating constant or a partition function. By proposing various forms for the internal energy function one can obtain suitable prior distributions.

All prior distributions including those of the previous sub-sections can be expressed as Gibbs distributions and we now consider some more which are strongly motivated by energy considerations.

Thin plate splines

The thin plate spline has found a great deal of applications recently in the image analysis literature (see [25] for a review). The thin plate spline transformation from the ideal template s_0 to a deformed version s has an energy function associated with it.

[6] introduced the thin-plate spline specifically for shape analysis of two-dimensional landmark data, although for a detailed account of thin-plate splines one should refer to [37] and [20]. The thin-plate spline is the most natural interpolating function for data in two dimensions and plays a similar role to the

natural cubic spline for interpolation in the one dimensional case. The natural cubic spline in one dimension is the unique interpolant $g(x)$ which minimises the roughness penalty

$$\int \frac{\partial^2 g}{\partial x^2} dx,$$

subject to interpolation at the 'knots'. The thin-plate spline minimises the 'bending energy' required to map s_0 to s, and the minimized bending energy is taken as the cost $E_{int}(s, s_0)$ following [27]. For an introduction to natural cubic splines (and thin plate splines in two dimensions) see [14]. Also note that the thin plate spline is a particular case of kriging (see, for example, [26] in this context).

4 The Intermediate Image G

Considering an intermediate image G in the method is appropriate for grey level templates. The deterministic approach is most often used. One possibility is to have an ideal 'average' grey level image and map this onto the deformed template by using a deformation. One appropriate mapping is the pair of thin-plate splines transformation mentioned in section 3.2.4 (see also [25]). Other transformation methods include the Fourier deformations of [1] and finite elements [23].

Alternative methods for specifying $G(S)$ include eigenfaces of [36] and [39] who use principal components of registered pixel grids. Also bivariate functions of pixel locations may be used to assign pixel levels in G. For example [34] use a bivariate normal density as a function to assign pixels in the central disc of a galaxy. This model represents decay of star light away from the disc centre.

5 Likelihood Models

When a generated image $G(S)$ is available this is often regarded as the mean image. Independent Gaussian noise is then added and this was the approach used by [1] and [25].

Alternatively consider the case when an intermediate image is not available. A texture model may be used to model behaviour inside and outside of the objects in the image. This is conveniently achieved using Markov/Gibbs random field models which are constructed from local properties of the image, but exhibit consistent global behaviour. These models have often been used in low level image analysis (eg. [3], [11]). This approach has been used for deformable templates by many authors including [18] and [32]. In general for an image F, a second order Gibbs random field takes the form

$$L(F) = \frac{1}{Z} \exp(-\frac{1}{2} \sum_i V_i(f(t_i)) + \sum_{i \sim j} V_{ij}(f(t_i), f(t_j))),$$

where the V_i are the single site potentials, the V_{ij} are the two site potentials and Z is the normalising constant. The sum of all the clique potentials is the energy function.

Here the notation $i \sim j$ is used to mean that pixel i is a neighbour of pixel j. In particular the conditional Gaussian MRF is simple having

$$V_i(f(t_i)) = (f(t_i) - \mu)^2/\tau^2,$$

$$V_{ij}(f(t_i), f(t_j)) = -\frac{\beta}{\tau^2}(f(t_i) - \mu)(f(t_j) - \mu).$$

Thus, the image model can be constructed as follows. Each region of the scene S has associated texture parameters μ_k, τ_k, β_k (for example inside a single template and outside the single template). The energy associated with the kth region is the sum of clique potentials and denoted by E_k. Hence, the likelihood or image model is given by

$$L(F \mid S) = \frac{1}{Z}\exp(-\frac{1}{2}\sum_k E_k).$$

The integrating constant is very complicated in general. Simulation from this distribution can by achieved using Markov Chain Monte Carlo (MCMC) schemes such as the Gibbs sampler and the Metropolis algorithm (eg. see [35]). Estimation of the mean parameters μ_k could be by grey level histogram inspection. Alternatively, all the unknown parameters could be estimated by maximum likelihood or treated as nuisance parameters throughout.

6 Inference

Inference about the scene is made through the posterior distributions of S, G obtained from equations (1)-(3). The full range of Bayesian statistical inference tools can be used and, as stated in section 2.1, the maximum of the posterior density (the MAP estimate) is most frequently used. Depending on the number of parameters, maximisation techniques such as a steepest ascent algorithm or simulated annealing could be straightforwardly used. Also, MCMC methods may be useful (eg. Smith and Roberts, 1993, Green, 1995 ; Besag et al, 1995) [35], [13], [4]) or approximations to the maxima such as Besag's [3] ICM could be used. There may be alternative occasions when particular template parameters are of great interest, in which case one would maximise the appropriate marginal posterior densities [32].

7 Many objects and context

Most of this work has been reviewed in terms of one template. However, a scene can be composed of a known number of templates or one multiple-object template with straightforward adaptation.

More difficult is the situation where an unknown number of objects are in the scene. The parameter space is then a mixture of discrete and continuous components and suitable techniques based on the Bayesian paradigm have been proposed by [18] (using jump diffusions), [2] (using spatial birth-death processes) and Green (1995) [13] (using reversible jump Markov chain Monte Carlo methodology). The computational issues are somewhat complicated but nevertheless can be dealt with in reasonable computational time.

By suitably specifying the prior model and including penalty terms in the likelihood, issues such as overlap of objects or non-allowable neighbouring objects

could be built into the procedure. Hence, the specification of high-level contextual information should be reasonably straightforward, although it will be very much application dependent. For example, [18] have built in penalties to prevent overlapping of mitochondria on the micrograph images. [33] have provided an 'integrated' approach for occluded multiple objects of different types.

Acknowledgements

I am grateful to Ian Dryden, John Kent and Alistair Walder for their comments; there are various ideas we have worked out jointly over the years and in particular this paper has been influenced from recent reviews and the grants for Gas board, EPSRC and BBSRC.

References

[1] Amit, Y., Grenander, U., and Piccioni, M. (1991). Structural image restoration through deformable templates. *Jour. Amer. Statist. Assoc.*, 86:376–387.

[2] Baddeley, A. J. and van Lieshout, M. N. M. (1994). Stochastic geometry models in high-level vision. In Mardia, K. V., editor, *Statistics and Images: Vol. 2*. Carfax, Oxford.

[3] Besag, J. E. (1986). On the statistical analysis of dirty pictures (with discussion). *Jour. Roy. Statist. Soc. B*, 48:259–302.

[4] Besag, J. E., Green, P, Higgden, D. and Mengerson, K. (1995). Bayesian computation and stochastic systems (with discussion). *Statistical Science*, **10**, pp3-66.

[5] Bookstein, F. L. (1986). Size and shape spaces for landmark data in two dimensions (with discussion). *Statist. Sci.*, 1:181–242.

[6] Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Machine Intell.*, 11:567–585.

[7] Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1992). Training models of shape from sets of examples. In Hogg, D. C. and Boyle, R. D., editors, *British Machine Vision Conference*, pages 9–18. Springer-Verlag.

[8] Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1994). Image search using flexible shape models generated from sets of examples. In Mardia, K. V., editor, *Statistics and Images: Vol. 2*. Carfax, Oxford.

[9] Dattatreya, G. R. (1991). Unsupervised context estimation in a mesh of pattern classes for image recognition. *Pattern Recognition*, 24:685–694.

[10] Dryden, I. L. and Mardia, K. V. (1991). General shape distributions in a plane. *Adv. Appl. Prob.*, 23:259–276.

[11] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741.

[12] Gong, P. and Howarth, P. J. (1992). Frequency based contextual classification and gray-level vector reduction for land-use identification. *Photogrammetric Engineering and Remote Sensing*, 58:423–437.

[13] Green, P. J. (1995). Overview: Monte Carlo methods. In Titterington, D. M., editor, *Complex Stochastic systems and Engineering*, pages 183–190. Clarendon Press, Oxford.

[14] Green, P. J. and Silverman, B. W. (1993). *Nonparametric regression and generalized linear models*, volume 58 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London.

[15] Grenander, U. (1993). *General Pattern Theory*. Clarendon Press, Oxford.

[16] Grenander, U., Chow, Y., and Keenan, D. M. (1991). *Hands: A pattern theoretic study of biological shapes*, volume 2 of *Research notes in neural computing*. Springer-Verlag, New York.

[17] Grenander, U. and Keenan, D. M. (1989). Towards automated image understanding. *Jour. Appl. Statist.*, 16:207–221.

[18] Grenander, U. and Miller, M. I. (1994). Representations of knowledge in complex systems (with discussion). *Jour. Roy. Statist. Soc. B:* , 56:285-299.

[19] Kent, J. T. (1994). The complex Bingham distribution and shape analysis. *Jour. Roy. Statist. Soc. B*, 56:285–299.

[20] Kent, J. T. and Mardia, K. V. (1994). The link between kriging and thin-plate splines. In Kelly, F. P., editor, *Probability, Statistics and Optimization*, pages 325–339. John Wiley and Sons.

[21] Kent, J.T., Mardia, K.V. and Walder, A.N. (1995). Conditional cyclic Markov random fields. *J. Appl. Prob.* In press.

[22] Knoerr, A. P. (1988). Global models of natural boundaries: theory and applications. Reports in Pattern Analysis 148, Division of Applied Mathematics, Brown University, Providence, RI.

[23] Marchant, J. A. (1993). Adding grey level information to point distribution models using finite elements. In *British Machine Vision Conference, 1993*, pages 309–318. BMVC Press.

[24] Mardia, K.V. (1994). *Statistics and Images.* (ed) Vol II Carfax Publishing Co. Abingdon.

[25] Mardia, K. V. and Hainsworth, T. J. (1993). Image warping and Bayesian reconstruction with grey-level templates. In Mardia, K. V. and Kanji, G. K., editors, *Statistics and Images: Vol. 1*, pages 257–280. Carfax, Oxford.

[26] Mardia, K. V., Kent, J. T., Goodall, C. R., and Little, J. A. (1995a). Kriging and splines with derivative information. To appear in *Biometrika*.

[27] Mardia, K. V., Kent, J. T., and Walder, A. N. (1991). Statistical shape models in image analysis. In Keramidas, E. M., editor, *Computer Science and Statistics: Proc. 23rd Symp. Interface*, pages 550–557. Interface Foundation, Fairfax Station.

[28] Mardia, K. V., Rabe, S., and Kent, J. T. (1995b). Statistics, shape and images. In Titterington, D. M., editor, *Complex Stochastic systems and Engineering*, pages 85–103. Clarendon Press, Oxford.

[29] Mardia, K.V. and Kanji, G.K. (eds) (1993). *Statistics and Images* Vol. I, 336pp Carfax Publishing Co., Abingdon.

[30] Mardia, K.V. and Gill, C.A. (eds) (1995). *Current Issues in Statistical Shape Analysis* Leeds University Press.

[31] Masson, P. and Pieczynski, W. (1993). SEM algorithm and unsupervised statistical segmentation of satellite images. *IEEE Trans. on Geoscience and Remote Sensing*, 31:618–633.

[32] Phillips, D. B. and Smith, A. F. M. (1994). Dynamic image analysis using Bayesian shape and texture models. In Mardia, K. V., editor, *Statistics and Images: Vol. 1*, pages 299–322. Carfax, Oxford.

[33] Qian, W. and Mardia, K. V. (1995). Recognition of multiple objects with occlusions. Technical Report 95/01, Dept. of Statistics, University of Leeds.

[34] Ripley, B. D. and Sutherland, A. I. (1990). Finding spiral structures in galaxies. *Phil. Trans. Roy. Soc. London A*, 332:477–485.

[35] Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion). *Jour. Roy. Statist. Soc. B*, 55:3–24.

[36] Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Jour. Cognitive Sci.*, 3:71–86.

[37] Wahba, G. (1990). *Spline models for observational data.* SIAM, Philadelphia.

[38] Yuille, A. L. (1991). Deformable templates for face recognition. *Jour. Cognitive Neuroscience*, 3:59–70.

[39] Yuille, A. L., Hallinan, P., and Cohen, D. (1992). Feature extraction from faces using deformable templates. *Int. Jour. Comp. Vision*, 8:99–111.

Cluster Monte Carlo Algorithms and their Applications

Jian-Sheng Wang

Computational Science Program,
National University of Singapore,
Lower Kent Ridge Road, Singapore 119260,
Republic of Singapore

cscwjs@leonis.nus.sg

Abstract. We introduce the Markov chain Monte Carlo method and review the background of the cluster algorithms in statistical physics. One of the first such successful algorithm was developed by Swendsen and Wang eight years ago. In contrast to the local algorithms, cluster algorithms update dynamical variables in a global fashion. Therefore, large changes are made in a single step. The method is very efficient, especially near the critical point of a second-order phase transition. Studies of various statistical mechanics models and some generalizations of the algorithm will be briefly reviewed. We mention applications in other fields, especially in imaging processing.

1 Monte Carlo Method

One of the important task in statistical mechanics and quantum field theory is to compute the statistical average,

$$\langle A \rangle = \frac{\sum_X A(X) \exp[-\beta H(X)]}{\sum_X \exp[-\beta H(X)]} = \sum_X A(X) P(X), \tag{1}$$

where the summation is over a set of all states X, which usually is very large. A general method was proposed long time ago in the fifties by Metropolis et al [1, 2], well-known as Metropolis algorithm. Instead of summing over all the states as required, one generates a selection of the states according to the probability $P(X)$, so that the answer is given approximately by an arithmetic average of the quantity $A(X)$. The generation of a sequence of the states is done stochastically by the method of Markov chain.

1.1 Markov chain

A Markov chain [3] is a mathematical notion for a random walk in the state space X. We start from some initial point X_0, the next point X_1 is generated according to certain transition probability. In this way a sequence of random points $X_0, X_1, \cdots, X_N$ is generated. These points are required to appear with

the probability $P(X)$. The rule to generate next point X_{i+1} given the point X_i is specified by the transition probability

$$P(X_{i+1}|X_i) = W(X_i \to X_{i+1}). \tag{2}$$

where $P(X_{i+1}|X_i)$ is the probability that the system is in state X_{i+1}, given that the system was in state X_i, namely, it is the conditional probability. We have

$$W(X_i \to X_{i+1}) \geq 0, \quad \sum_{X_{i+1}} W(X_i \to X_{i+1}) = 1. \tag{3}$$

Since W is probability, it satisfies the usual constraint of probability—probability must be non-negative and the total probability is 1.

If we choose W so that

$$P(X) = \sum_{X'} P(X')W(X' \to X), \tag{4}$$

we say that $P(X)$ is invariant under the transition of W, or $P(X)$ is the equilibrium distribution for the given transition probability. Such a probability distribution will be unique if starting from any state X, it is possible to get to any other state X' after a finite number of transitions. We call such a Markov chain ergodic (regular, or irreducible).

1.2 Detailed balance

How to choose W to get a prescribed $P(X)$? It is sufficient that W satisfies

$$P(X')W(X' \to X) = P(X)W(X \to X'), \tag{5}$$

where $P(X)$ is the known distribution. If W satisfies this equation and it is ergodic, then the Markov chain will generate X according to the probability distribution $P(X)$.

1.3 Metropolis algorithm

Metropolis method goes as follows: starting from some initial state (configuration) X_0, a sequence of states will be generated by a Markov chain through the transition probability $W(X_i \to X_{i+1})$. The starting state may be a unique state or may be generated at random with certain distribution P_0. The new state X_{i+1} is based on the old state X_i. The new state is generated in such a way so that the probability that the system is in state X_{i+1} is just $W(X_i \to X_{i+1})$, knowing that the system was in state X_i. Hopefully we can generate this probability distribution easily. Otherwise we might have generated $P(X)$ directly in the first place. It is sufficient that W satisfies detailed balance condition and the Markov chain is ergodic. Metropolis algorithm refers to a specific choice of W:

$$W(X \to X') = T(X \to X') \min\left(1, \frac{P(X')}{P(X)}\right), \tag{6}$$

where $X \neq X'$ and $T(X \to X') = T(X' \to X)$. Matrix T can be any distribution but must be symmetric. We almost always use

$$T(X \to X') = \begin{cases} const, & \text{for } X' \text{ in a region around } X; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The probability that the state does not change, $W(X \to X)$, is determined by the normalization condition

$$\sum_{X'} W(X \to X') = 1. \tag{8}$$

2 Ising Model for Magnets

To be able to understand Swendsen-Wang and other cluster algorithms, we need to introduce the very popular model for ferromagnets in condensed matter physics. Ising model [4, 5] is the simplest model for ferromagnets. One of the striking feature of a magnet is that it has spontaneous magnetization at lower temperatures. However, the magnetization disappears at some higher temperature T_c. The phase below T_c is called ferromagnetic phase and that above T_c is called paramagnetic phase. T_c is called critical temperature and there is a phase transition at this temperature.

Let's take two-dimensional system as an example. On an $L \times L$ square lattice, we put a spin σ_i at each lattice site i. Each spin takes only two possible values $+1$ and -1. The set of L^2 spins consists of the state space X. The system is specified completely by the energy function,

$$H(\{\sigma\}) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{9}$$

where J is called coupling constant. The summation is over nearest neighbors only, i.e., each bond is summed once. Such model is also referred as nearest neighbor Ising model. When J is positive, the model describes a ferromagnet; while $J < 0$ describes an anti-ferromagnet. Let's consider just one term in the nearest neighbor interaction. The interaction energy of neighbors takes only two values, $+J$ or $-J$. For ferromagnet, the energy is lower if the two spins are parallel. It costs energy if the spins are pointing in different directions.

According to statistical-mechanical description, the system will not be in a definite state. Due to thermal fluctuation, the system will have some probability distribution among the states. If the temperature T is a fixed parameter, we have the famous Boltzmann distribution

$$P(\{\sigma\}) \propto \exp\left(-\frac{H(\{\sigma\})}{k_B T}\right), \tag{10}$$

where $k_B \approx 1.38 \times 10^{-23}$ Joules/Kelvin is called the Boltzmann constant. Our goal is to calculate average quantities such as energy, magnetization, correlation

functions, etc. For example, the internal energy is computed as

$$U = \langle H \rangle = \sum_{\{\sigma\}} H(\{\sigma\}) P(\{\sigma\}). \tag{11}$$

The summation is over all possible states. Since each site can have two states (spin up and spin down), we have 2^{L^d} states for a system of linear size L in d dimensions.

A Monte Carlo simulation of the Ising model by Metropolis algorithm involves the following steps:

1. Initialize σ_i at each site with arbitrary spin values, e.g., all spin up, or up or down with equal probability. The complete set of spins $\{\sigma\}$ is the abstract state X discussed in the previous section.
2. Choose a site k at random and propose to flip the spin at that site, $\sigma'_k = -\sigma_k$. The proposed state $\overline{X}$ is the one with one spin at location k flipped. This amounts to take $T(X \to \overline{X})$ equal to $1/L^d$ if X and $\overline{X}$ differ by 1 spin, and 0 otherwise.
3. Calculate the energy increment

$$\Delta E = H(\{\sigma'\}) - H(\{\sigma\}) = 2J\sigma_k \sum_{\substack{\text{neighbors} \\ \text{of } k}} \sigma_i. \tag{12}$$

4. Accept the proposed state as the new state if a uniformly distributed random number (between 0 and 1) is less than $\exp(-\Delta E/k_B T)$; retain the old state as the new state otherwise.
5. go to 2.

One Monte Carlo step will be defined as performing the above basic single-spin flip L^d number of times. After one Monte Carlo step, each site on average has tried once to flip.

3 Critical Slowing Down

In Ising model simulation, or more generally any simulation using the Metropolis algorithm, the next configuration depends on the previous one. Due to this correlation between Monte Carlo steps, the formula for the statistical error $\epsilon \approx \sigma/\sqrt{N}$ underestimates the true error, where σ^2 is the variance of the quantity of interest, and N is Monte Carlo steps. The error formula should be replaced by

$$\epsilon \approx \sigma \sqrt{\frac{1+2\tau}{N}}. \tag{13}$$

That is, the error is larger by a factor $\sqrt{1+2\tau}$ than the uncorrelated data. The quantity τ is called correlation time, which is roughly the number of Monte

Carlo steps needed to generate independent configurations. We can compute the correlation time τ as follows: define the time-dependent correlation function,

$$f(t) = \frac{\langle A(t')A(t'+t)\rangle - \langle A\rangle^2}{\langle A^2\rangle - \langle A\rangle^2}, \tag{14}$$

where time t is measured in terms of Monte Carlo steps—time is passed by one unit when one updates the system by one Monte Carlo step. And $A(t')$ is the quantity at steps t' whose error we want to estimate. The angular brackets denote average over Monte Carlo steps. The correlation time is defined by

$$\tau = \sum_{i=1}^{\infty} f(t). \tag{15}$$

Now we are ready to explain the concept of critical slowing down. Just like many other quantities near the critical point, the correlation time also becomes singular at the critical point. On an infinite lattice, it diverges with a power law

$$\tau \propto |T - T_c|^{-\nu z}. \tag{16}$$

Here ν is the correlation length exponent and z is called dynamic critical exponent. This phenomenon that the correlation time becomes very large is called critical slowing down. It not only happens in computer simulation but also appears in real systems. On a finite system of linear size L, the correlation time will, of course, not go to infinity. But it will grow with size as

$$\tau \propto L^z, \quad T = T_c. \tag{17}$$

Substituting this result into the error formula, we find that $\epsilon \approx \sigma L^{z/2}/N^{1/2}$ for large system near T_c. For the two-dimensional Ising model with Metropolis algorithm, $z \approx 2.1$, we see typically that increasing the size leads to a larger error in the quantity to be calculated.

4 Cluster Monte Carlo Algorithms

The Metropolis algorithm makes changes locally one site at a time. This is the cause of the critical slowing down. There are algorithms that change the states by a group of clusters. The advantage of the cluster algorithms [6] is that they are faster, not necessarily in computer time per Monte Carlo step, but in terms of dynamics. That is, it has a much smaller value of z.

4.1 Swendsen-Wang algorithm

The Swendsen-Wang multi-cluster algorithm [7] is closely related to percolation [8] based on the work of Fortuin and Kasteleyn [9]. One starts with a spin configuration $\{\sigma\}$ and generates a percolation configuration based on the spin configuration by the rule described below. Then the old spin configuration is

forgotten and a new spin configuration $\{\sigma'\}$ is generated based on the percolation configuration. The rule is such that the detailed balance condition is satisfied. Or more generally, the transition leaves the equilibrium probability invariant. The algorithm goes as follows (for ferromagnet $J > 0$):

1. Start with some arbitrary state $\{\sigma\}$.
2. Go through each nearest neighbor connection of the lattice, create a bond between the two neighboring sites i and j with probability $p = 1 - e^{-2J/k_BT}$, but only if the spins are the same, $\sigma_i = \sigma_j$. Never put a bond between the sites if the spin values are different. We are creating a bond percolation configuration with probability p on a subset of lattice sites where the spins are either all pointing up or down.
3. Identify clusters as a set of sites connected by bonds, or isolated sites. Two sites are said to be in the same cluster if there is a connected path of bonds joining them. Every site has to belong to one of the clusters. After the clusters are found, each cluster is assigned a new Ising spin chosen with equal probability between $+1$ and -1. The old spin values now can be 'forgotten'. And all the sites in a cluster now take the value of spin assigned to the cluster.
4. One Monte Carlo step is finished. Repeat 2 for the next step.

The performance of the algorithm in terms of correlation time in comparison with Metropolis algorithm is remarkable. Recall that the dynamic critical exponent z is about 2 for Metropolis algorithm almost independent of the dimensionality. The Swendsen-Wang algorithm in one dimension gives $z = 0$ (τ approaches a constant as $T \to 0$). For the two-dimensional Ising model, the dynamic critical exponent z is less than 0.3 or possibly zero (but $\tau \propto \ln L$). In three dimensions it is about 0.5. At and above four dimensions it is 1.

4.2 Wolff single-cluster algorithm

In the Swendsen-Wang algorithm, we generated many clusters and then flipped these clusters. Wolff algorithm [10] is a variation on the way clusters are flipped. One picks a site at random, and then generates one single cluster by growing a cluster from the seed. The neighbors of the seed site will also belong to the cluster if the spins are in parallel and a random number is less than $p = 1 - e^{-2J/k_BT}$. That is, the neighboring site will be in the same cluster as the seed site with probability p if the spins have the same sign. If the spins are different, neighboring site will never belong to the cluster. Neighbors of each new site in the cluster are tested for membership. This testing of membership is performed on pair of sites (forming a nearest neighbor bond) not more than once. The recursive process will eventually terminate. The spins in the cluster are turned over with probability 1. Single cluster algorithm appears more efficient than the multi-cluster one.

The following is a fairly elegant way of implementing the Wolff algorithm in C. The function is the core part which performs a Wolff single cluster flip. This function is recursive. The array for spins `s[ ]`, percolation probability `p`, and

coordination number Z (the number of neighbors) are passed globally. The first argument i of the flip function is the site to be flipped, the second argument s0 is the spin of the cluster before flipping. The function neighbor returns an array of neighbors of the current site. The function drand48() returns a random number unformly distributed between 0 and 1.

```
void flip(int i, int s0)
{
   int j, nn[Z];
   s[i] = - s0;                                flip the spin immediately
   neighbor(i, nn);                            find nearest neighbors of i
   for(j = 0; j < Z; ++j)                      flip the neighbor if
      if(s0 == s[nn[j]] && drand48() < p)      spins are equal and
         flip(nn[j], s0);                      random number is smaller than p
}
```

5 Further Development of Cluster Algorithms

The attractive feature of Swendsen-Wang algorithm stimulated research for other types of cluster algorithms and for its theory. The original method works for Ising and Potts models. It is not clear at the first sight how one can generalized it for arbitrary model. The idea of Ising spin embedding into a model appears to work well. This idea is applied to a number of models, including models with continuous degree of freedom [10, 11, 12]. Kandel and coworkers proposed a general scheme for cluster algorithms [13, 14], which leads to efficient algorithms for models with competing interactions [15, 16]. Cluster algorithm of a different kind (loop algorithm) was proposed to work with vertex models [17].

Sokal's idea of introducing auxiliary variables [18] is another way of generalizing the cluster algorithms and is drawing attention to statisticians [19]. However, this method seems not work well in comparison with embedding algorithms when the system has Ising symmetry and embedding can be implemented.

A large class of problems in statistical mechanics can be now simulated with cluster algorithms. Cluster algorithms are used routinely where traditionally local algorithms were used. Due to space limitations, this types of applications will not be reviewed here.

6 Application in Image Processing

The Swendsen-Wang algorithm and some generalizations were used in connection with image processing [19]. Random images [20], in particular, a class of random fields called Markov random fields, correspond closely to systems studied in statistical physics. Thus the Monte Carlo techniques developed in statistical mechanics can be readily used here. There is also the need to estimate model parameters [21], this can be done by Monte Carlo simulation. The most important application appears to be in the problems of image restoration [22] and image

segmentation [23, 24]. The statistical method (or Bayesian inference) in image analysis is relatively new. The basic problem in Bayesian image restoration is the following. Let X be the true image, which is also statistical in nature and distributed according to $P(X)$, called the prior. The true scene, X, is not directly observable, and the observed data Y contains noise. Given Y, one wants to compute the distribution of X, namely, the conditional probability $P(X|Y)$, called posterior. This is obtained by noting

$$P(X|Y) \propto P(Y|X)P(X), \qquad \text{for fixed } Y. \tag{18}$$

Image restoration amounts to find the most probable X by simulating the probability distribution $P(X|Y)$. Ising model is assumed for $P(X)$ in some application; while $P(Y|X)$ is given as independent Gaussian distribution for Y. From the point of view of statistical mechanics, $P(X|Y)$ describes a model with inhomogeneous local magnetic field which is determined by Y.

Segmentation is a process of classifying each pixel in an image. We can think of it as a special case of restoration problem. Thus the two problems share the same mathematical structure. The problem of speed of convergence in image segmentation by Swendsen-Wang algorithm is studied in [23] and [24]. The cluster algorithms work best when the system has a high degree of symmetry. The two types of problems, image restoration and image segmentation, all end up involving the simulation of a model with the presence of complicated magnetic field, which makes cluster algorithms less effective. The cluster algorithms and other acceleration algorithms, e.g., ref. [25], developed in statistical physics may be very helpful in this field. But much more work needs to be done.

Acknowledgements

I thank Dr. S. Z. Li for inviting me for this talk, and Prof. K. V. Mardia for drawing me the attending of ref [19].

References

1. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, Vol. 21, pp. 1087-1092, 1953.
2. M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, Volume I: Basics, John Wiley & Sons, New York, 1986.
3. L. E. Reichl, *A modern Course in Statistical Physics*, Edward Arnold, 1980.
4. E. Ising, "Beitrag zur theorie des ferromagnetismus," *Z. Phys.*, Vol. 31, pp. 253, 1925.
5. L. Onsager, "Crystal statistics I: A two-dimensional model with an order-disorder transition," *Phys. Rev.*, Vol. 65, pp. 117-149, 1944.
6. For reviews, see A. D. Sokal, "Monte Carlo methods in statistical mechanics: foundations and new algorithms," *Cours de Troisième Cycle de la Physique en Suisse Romande*, Lausanne, 1989; J.-S. Wang and R. H. Swendsen, "Cluster Monte

Carlo algorithms," *Physica A*, Vol. 167, pp. 565-579, 1990; R. H. Swendsen, J.-S. Wang, and A. M. Ferrenberg, "New Monte Carlo methods for improved efficiency of computer simulation in statistical mechanics," in *The Monte Carlo Method in Condensed Matter Physics*, ed. K. Binder, Topics in Applied Physics, Vol. 71, pp. 75-91, Springer, Berlin, 1992.

7. R. H. Swendsen and J.-S. Wang, "Non-universal critical dynamics in Monte Carlo simulations," *Phys. Rev. Lett.*, Vol. 58, pp. 86–88, 1987.
8. D. Stauffer and A. Aharony, "Introduction to Percolation Theory," 2nd ed, Taylor & Francis, London, 1992.
9. C. M. Fortuin and P. W. Kasteleyn, "On the random cluster model. I. Introduction and relation to other models," *Physica (Utrecht)* Vol. 57, pp. 536-564, 1972.
10. U. Wolff, "Collective Monte Carlo updating for spin systems," *Phys. Rev. Lett.*, Vol. 62, pp. 361-364, 1989.
11. R. Brower, "Embedded dynamics for ϕ^4 theory," *Phys. Rev. Lett.*, Vol. 62, pp. 1087-1090, 1989.
12. J.-S. Wang, R. H. Swendsen, and R. Kotecký, "Antiferromagnetic Potts models," *Phys. Rev. Lett.* Vol. 63, pp. 109-112, 1989.
13. D. Kandel, E. Domany, D. Ron, and A Brandt, "Simulations without critical slowing down", *Phys. Rev. Lett.*, Vol. 60, pp. 1591-1594, 1988.
14. D. Kandel, E. Domany, and A. Brandt, "Simulations without critical slowing down: Ising and three-state Potts models," *Phys. Rev. B*, vol. 40, pp. 330-344, 1989.
15. D. Kandel, R. Ben-Av, and E. Domany, "Cluster dynamics for fully frustrated systems," *Phys. Rev. Lett*, Vol. 65, pp. 941-944, 1990.
16. S. Liang, "Application of cluster algorithms to spin glasses," *Phys. Rev. Lett.*, Vol. 69, pp. 2145-2148, 1992.
17. H. G. Evertz, G. Lana, and M. Marcu, "Cluster algorithms for vertex models," *Phys. Rev. Lett.*, Vol. 70, pp. 875-879, 1993.
18. R. G. Edwards and A. D. Sokal, "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm," *Phys. Rev. D*, Vol. 38, pp. 2009-2012, 1988.
19. J. Besag and P. J. Green, "Spatial statistics and Bayesian computation," *J. R. Statist. Soc. B*, Vol. 55, pp. 25-37 and 53-102, 1993.
20. A. J. Gray, J. W. Kay, and D. M. Titterington, "An empirical study of the simulation of various models used for images," *IEEE Trans. Pattern Anal. Machine Intell.*," Vol. 16, pp. 507-513, 1994.
21. G. G. Potamianos and J. K. Goutsias, "Partition function estimation of Gibbs random field images using Monte Carlo simulations," *IEEE Trans. Inform. Theory*, Vol. 39, pp. 1322-1332, 1993.
22. A. J. Gray, "Simulating posterior Gibbs distributions: a comparison of the Swendsen-Wang and Gibbs sampler methods," *Statistics and Computing*, Vol. 4, pp. 189-201, 1994.
23. I. Gaudron, "Vitesse de convergence de la dynamique de Swendsen-Wang pour minimiser des énergies de segmentation d'images," *C. R. Acad. Sci. Paris*, t. 320, Série I, pp. 475, 1995.
24. I. Gaudron, "Rate of convergence of the Swendsen-Wang dynamics in image segmentation problems: A theoretical and experimental study," *J. Stat. Phys.*, 1996.
25. M. A. Novotny, "Monte Carlo algorithms with absorbing Markov Chains: Fast local algorithms for slow dynamics," *Phys. Rev. Lett.*, Vol. 74, pp. 1-5, 1995.

Image Estimation

Gibbs Sampler and Maximum Likelihood Estimation for Unsupervised Image Segmentations

Chee Sun Won
Dept. of Electronics Engineering
Dongguk University
Seoul, 100-715, Korea
cswon@dgu4680.dongguk.ac.kr

Abstract

In this paper, we propose a new way of obtaining an unsupervised image segmentation. Similar to the previous unsupervised image segmentation algorithms, the proposed algorithm is iterative. However, we adopt a different stopping rule such that if the region label at each pixel does not change for a sufficient number of iterations, then we confirm it as a final one and never visit the corresponding pixel for the update thereafter. Eventually, all pixels in the image will be confirmed to assure the convergence. Another feature of the proposed algorithm is that the Gibbs sampler is adopted to generate a region label for the update. This probabilistic region label update will resolve the problem of sticking at local maximum too early. Adding all these features together, the proposed algorithm updates region labels with Gibbs sampler using the current model parameter values and then, using the current region labels, the maximum likelihood estimates of the model parameters are obtained. This alternate procedure continues until all pixels in the image are declared as the confirmed final labels. Experimental results on the synthesized images with independent Gaussian noise show that the proposed algorithm yields significantly lower error rates than the previous deterministic unsupervised segmentation algorithms.

1. Introduction

Unsupervised image segmentation is to estimate the model parameter values during the segmentation procedure. Depending on how to incorporate model parameter estimation into the region label updates, one can devise various unsupervised segmentation algorithms. For example, in an unsupervised version of the ICM (denoted as UICM in this paper) [1], given the current region labels, the model parameters are estimated and then, using the estimated model parameters, the region label at each pixel is updated. These two steps are repeated until a convergence condition is satisfied. Therefore, the region label update and the model parameter estimate are executed in an alternate fashion. In the Won and Derin's (WD) [2] algorithm, however, the closed form expressed ML estimates of the noisy model parameters are plugged into the MAP

This work was supported by KOSEF-1995.

criterion to be an objective function with respect to the region labels only. Therefore, whenever the region label is updated to increase the objective function, it reflects the effect of the ML estimates of the noisy image model parameters into the segmentation. Then, the resulting unsupervised segmentation turns out to be a simultaneous parameter estimation and segmentation procedure. The Zhang, Modestino, and Langan's (ZML) [3] algorithm for the independent Gaussian noisy image is similar to the UICM algorithm in that it alternately executes the region label updates and the ML estimate of the model parameters. The difference of the two algorithms is that the region label at each pixel of the ZML algorithm takes a probabilistic "soft" decision, whereas in the UICM (and also in the WD) a deterministic "hard" decision is made. Consequently, although the above three algorithms are all seeking a local maximum for the criterion obtained from the same stochastic models, the speed of convergence to a local maximum and the final error rate may be different from each other.

In [4], convergence properties of the three unsupervised segmentation algorithms were investigated. Among the three algorithms, the WD algorithm showed the fastest convergence property, whereas the ZML algorithm yielded the lowest error rates. Note that the ZML algorithm showed an intermediate characteristics between the deterministic relaxation algorithm and the stochastic one such as the simulated annealing. That is, unlike the deterministic relaxation algorithms the ZML algorithm updates the region label with a probability to avoid the the local maxima during the iterations. However, it does not guarantee the convergence to the global maximum as opposed to the stochastic relaxation methods.

The goal of this paper is to propose a new probabilistic unsupervised image segmentation algorithm that has the similar convergence property to the ZML algorithm, but with less memory size and faster execution time.

2. Image Models and Motivations

Suppose that the observed image is a realization $y=\{y_t : t \in \Omega\}$ from a random field $Y=\{Y_t : t \in \Omega\}$, where Ω is a discrete rectangular lattice denoted as $\Omega = \{(i,j): 1 \le i \le N_1, 1 \le j \le N_2\}$. For each pixel $t \in \Omega$, the region-type that the pixel belongs to is specified by a region label. The region label at a pixel t is modelled as a discrete-valued random variable X_t taking one of values in $\{1,\cdots,G\}$. So, the set of region labels $x = \{x_t, t \in \Omega\}$ is a realization of a random field for a region label process $X = \{X_t, t \in \Omega\}$.

The gray level r_g with a region label $x_t=g$ at $t \in \Omega$ is assumed to be corrupted by an independent Gaussian noise. That is, at $t \in \Omega$, y_t is

Gaussian with mean r_g and and variance σ^2 for $x_t = g \in \{1,\cdots,G\}$. Then,

$$P(Y|X) = \prod_{t\in\Omega} P(Y_t|\theta_Y, x_t) \tag{1}$$

where

$$P(Y_t = y_t|\theta_Y, x_t = g) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{(y_t - r_g)^2}{2\sigma^2}}, \tag{2}$$

and $\theta_Y = \{\sigma^2, r_1, \cdots, r_G\}$ represents the set of noisy image model parameters associated with Y. The region process is assumed to be a MRF with respect to a neighborhood system $\eta_t = \{\eta_t; t\in\Omega\}$ and is characterized by the local characteristics;

$$P(X_t = x_t|X_q = x_q, q\in\Omega, q\neq t) = P(X_t = x_t|X_{\eta_t} = x_{\eta_t}) \tag{3}$$

where η_t is the neighborhood of $t\in\Omega$ such that $t\notin\eta_t$ and $q\in\eta_t$ implies $t\in\eta_q$. Exploiting the MRF-GRF equivalence, we can specify the region label process X through its joint distribution. For example, adopting the MLL(Multi-level logistic) distribution[5,6] with the pair-clique potential for the second order neighborhood system, the joint distribution of the region process X can be represented as follows

$$P(X = x') = \frac{e^{\beta v(x')}}{\sum_x e^{\beta v(x)}}, \tag{4}$$

where $v(x')$ can be obtained by subtracting the number of neighbor pairs having unequal region labels from that of equal ones and β is the corresponding pair-clique parameter. Note that the segmentation results are relatively insensitive to the choice of β value if it is selected from a certain range of values[2,3]. Therefore, the β value is assumed to be known and is fixed in this paper. Now, the unsupervised image segmentation is to obtain the optimal region label and the estimated noisy model parameter set (x^*, θ_Y^*) from the noisy image data y. To obtain the optimal set, the deterministic unsupervised segmentation algorithms typically take the following two steps :

Obtain an initial region label estimate
<u>REPEAT</u>

A. Obtain an ML estimate of θ_Y

B. Update the region labels by maximizing local conditional probabilities

UNTIL (the change of the region labels is less than a threshold or the number of iteration reaches the predetermined value)

The UICM and WD algorithm basically follow the above two steps. In the WD algorithm, however, since the closed-form expressed ML estimates of θ_Y are plugged into the original objective function, it effectively merges step A with B. That is, everytime a region label is examined for the update, the corresponding ML estimates of the model parameters are implicitly considered. Despite the difference of the model parameter estimations, the two algorithms are classified into the same group because of their deterministic region label updates (i. e., maximization of a function to obtain the updated region label). Also, because of the deterministic way of choosing the region labels, both algorithms yield a quick convergence [4].

Another unsupervised segmentation is to make use of the EM procedure. In the ZML algorithm [3], the computationally difficult conditional expectation in the EM procedure was obtained by using the pseudo-likelihood approximation as in [1]. Then, the unsupervised segmentation algorithm based on the EM procedure for the independent Gaussian noise can be obtained by the following steps [3,4] :

Obtain an initial region label estimate

REPEAT

A. Obtain a fuzzy ML estimate of θ_Y

B. Obtain the local conditional probabilities (conditional expectations) for all region labels at each pixel

UNTIL (the changes of the parameter values are less than a threshold or the number of iterations reaches the predetermined value)

At each pixel $t \in \Omega$, assign the final region label as

$$g_t^* = \underset{g \in \{1, \cdots, G\}}{\arg\max} \ \hat{x}_{tg}.$$

Note that given the model assumptions the conditional expectations of the region labels are equivalent to the calculations of the local conditional probabilities [4]. As a result of taking the soft (probabilistic) decisions, the ZML algorithm never fall in the local maximum during the iterations. So, as expected, the ZML algorithm, although the speed of the initial convergence is slow, yields a lower final error rate [4]. Therefore, it can be viewed that the ZML algorithm resides in somewhat between the

deterministic relaxation and the stochastic relaxation algorithms. Although the ZML never guarantees the global convergence, it takes shorter execution time to the convergence than the simulated annealing. A drawback of the ZML algorithm is that it needs a larger memory size to store all possible soft decisions at each pixel. Motivated by this fact, in this paper, we propose a probabilistic relaxation algorithm, which yields a similar error rate to the ZML algorithm with a smaller memory size and faster convergence.

3. Proposed Algorithm

A drawback of the deterministic unsupervised segmentation algorithm is that it is unduly sensitive to the nearby local maximum. To alleviate this problem, let us consider the following modification:

Obtain an initial region label estimate
<u>REPEAT</u>
A. ML estimate of θ_Y
B. Region label update by adopting the Gibbs sampler at each pixel (i. e., obtain a sample from the local characteristics for each pixel)
<u>UNTIL</u> (the change of the region labels is less than a threshold or the number of iteration reaches the predetermined value)

Note that the above algorithm also consists of two steps. In step B, however, given the current model parameter values, the region label is updated by the Gibbs sampler (rather than the maximization procedure). Note that if the model parameter θ_Y is known (i. e., without the step A), the above algorithm, as the iteration goes to infinite, yields a realization from the Gibbs distribution [7]. More specifically, from the assumed stochastic image models, we have

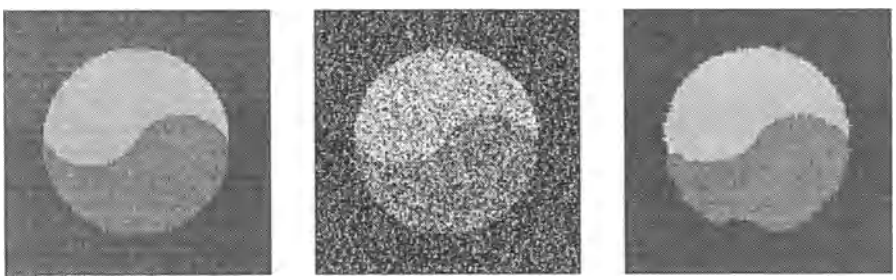
$$P(X \mid Y) \triangleq \pi(x) = \frac{1}{Z} e^{U^p(x)} \tag{5}$$

$$\text{where } U^p(x) = \beta v(x) - \frac{\sum_{t \in \Omega} (y_t - r_{x_t})^2}{2\sigma^2} \tag{6}$$

$$\text{and } Z = \sum_x e^{U^p(x)} \tag{7}$$

Then, starting from an arbitrary initial realization, the Gibbs sampler yields a realization from $\pi(x)$ for the infinite number of iterations [7]. In other words, as we sample a region label from the local characteristics

by visiting each pixel infinitely often, the final region label configuration will be a sample from the joint Gibbs distribution $\pi(x)$. On the other hand, with the step A, it is equivalent to interrupt the sampling procedure to estimate the model parameters with the current samples. Note that this is similar to the simultaneous segmentation algorithm proposed by Lakshmanan and Derin [8]. However, there are following three differences. First, in [8], instead of θ_Y, the region model parameter θ_X was estimated at step A (θ_Y was assumed to be known). Second, at step B, they visit the same pixel sufficiently often, whereas the above algorithm visits each pixel only once for the current estimated model parameters. Third, since their algorithm is based on the simulated annealing, a cooling schedule was adopted.



a) Original Image b) Noisy Image (SNR=1) c) Gibbs Sampled Image (at 200th iteration)

Fig. 1. A Gibbs sample of the 3-region image (starting from an arbitrary initial region label realization)

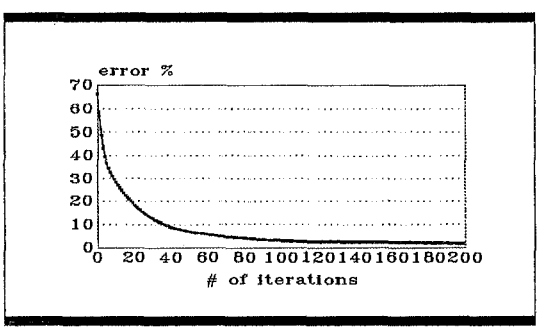


Fig. 2. Error rate of the Gibbs sampler for Fig.1

Although the convergence of the above algorithm has not been formally proved yet, the experimental results shown in figures 1 and 2 justify the convergence of the proposed algorithm. That is, with an arbitrary initial region labels, the above algorithm consistently reduces the error rate as the iteration increases. It is conjectured that the constraint on the region label continuity (i. e., the known β) guides the convergence at the initial stages of the iterations and then for the subsequent iterations the estimated model parameter $\hat{\theta}_Y$ and the region label continuity β together accelerate the convergence. Note that due to the probabilistic updates of the Gibbs sampler, the algorithm is less vulnerable to the local maximum. That is, since the region labels are updated with a probability, the algorithm does not strictly increase the objective function. Rather, the region labels with a higher probability will have a higher chance to be selected for the update. However, as a result of the probabilistic selection, there will be a fluctuation in the error rates near convergence. In order to avoid this problem and to further accelerate the convergence we impose a constraint on the stopping rule such that if the region label at each pixel does not change for a sufficient number of iterations, then we confirm it as a final region label and never visit the corresponding pixel thereafter. Eventually, all pixels in the image will be confirmed to assure the convergence of the the proposed iterative algorithm. Now the proposed algorithm can be stated as follows :

```
Let pixel_flag(i,j)=0 for all (i,j)∈Ω
Let counter(i,j)=0 for all (i,j)∈Ω
iter =0, set threshold1 and threshold2
Obtain an initial region label estimate
REPEAT
 A. Obtain an ML estimate of θ_Y
 B. REPEAT
      IF (iter >= threshold1) THEN
        IF (pixel_flag(i,j) = 0) THEN
           update x(i,j) via the Gibbs sampler
           IF (counter(i,j) > threshold2) THEN pixel_flag(i,j)=1
           ELSE IF (previous region label = updated region label) THEN
                      counter(i,j)=counter(i,j)+1
                 ELSE counter(i,j)=0
           ENDIF
        ENDIF
    ELSE
        update x(i,j) via the Gibbs sampler
    ENDIF
```

```
    UNTIL (visiting all pixels (i,j)∈Ω)
    iter=iter+1
    IF (iter >= threshold1 ) THEN threshold2=threshold2-1
    ENDIF
UNTIL (pixel_flag(i,j)=1 for all (i,j)∈Ω)
```

Once the number of iterations exceeds threshold1, at each pixel, we accumulate the number of subsequent iterations for which the region label is unchanged. If the accumulated count exceeds the threshold2, we flagged the pixel as the converged one. Since the threshold2 decreases as the iteration increases, all pixels will be flagged eventually. The proposed constraint for the pixel visit can be interpreted as a "spatial annealing" procedure. Note that in [7,8] the region label updating procedure was annealed by reducing the temperature in the energy function. Here, we annealed the procedure by skipping pixels with almost-fixed region labels. Both annealing schemes provide the stopping rule for the iterative region label updates. However, unlike the proposed annealing scheme, the annealing procedure in [7] guarantees the convergence to the global maximum. Instead, the "spatial annealing" procedure removes the fluctuations of the region labels at convergence. Moreover, since we skip pixels with almost-fixed region labels, the execution time for the region label update will be reduced as the iteration increases.

4. Experiments

To demonstrate the convergence properties of the proposed algorithm, we compare the segmentation results with those of the UICM (the sequential version [4]) and the ZML algorithms. To be fair, we used the same initial region labels for all algorithms. For the initial region label segmentation, we used the uniform quantizer method. So, dividing the difference between the highest and lowest gray levels into the number of regions uniformly, we assign the initial region label which is closest to the quantized level. Moreover, although the region labels of the ZML algorithm are decided only after the final iteration, to compare the speed of the convergence, we also obtain the region labels that maximize the current local conditional probabilities for the ZML algorithm at each iteration. Assuming $\beta = 0.8$, we obtained the segmented results for 2-, 3-, and 4-region synthesized and Gibbsian images. They all showed similar convergence properties. Among them, for a 4-region synthesized image, the segmented result and the error rate for each iteration are shown in figures 3 and 4, respectively. Considering all the experimental results, we can acknowledge the following facts:

1) The ZML and the proposed algorithm yield much lower error rates than the deterministic UICM.
2) There is no observed dominance in terms of the final error rates between the ZML and the proposed algorithm.
3) To store the region label information during the iteration, the proposed algorithm requires only N_1N_2 integer memory units, whereas the ZML needs GN_1N_2 ones.
4) Since we skip the almost converged pixels during the iterations, the execution time of the proposed algorithm is shorter than that of the ZML algorithm.

5. Conclusion

In this paper, we propose to use the Gibbs sampler for the region label update of the iterative unsupervised image segmentation algorithm. Since the Gibbs sampler chooses the region label with a probability, it is less vulnerable to the local maximum. Moreover, since the proposed algorithm adopts an "spatial annealing" scheme by skipping pixels with almost-fixed region labels, it will accelerate the speed of the convergence.

References

1. J. E. Besag, "On the statistical analysis of dirty pictures," J.R.Statist. Soc., Ser. B, vol.48, pp.259-302, 1986.
2. C. S. Won and H. Derin, "Unsupervised segmentation of noisy and textured images using Markov random fields," CVGIP:Graphical Models and Image Processing, vol.54(4), pp.308-328, 1992.
3. J. Zhang, J. W. Modestino, and D. A. Langan, "Maximum-likelihood parameter estimation for unsupervised stochastic model-based image segmentation," IEEE Trans. on Image Processing, vol.3, no.4, pp.404-220, July 1994.
4. C.S.Won, "Convergence of unsupervised image segmentation algorithms," SPIE proceedings, vol.2568, pp.209-220, 1995.
5. H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random field," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.PAMI-9, pp.39-55, 1987.
6. A.K. Jain and S.G. Nadabar, "Markov random field applications in image analysis," Data Analysis in Astronomy IV, V. Di Gesu, et.al. (Eds.), pp.39-50, 1992.
7. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Baysian restoration of images," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.PAMI-6 (6), pp.721-741, 1984.

8. S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields," IEEE Trans. Pattern Anal. Mach. Intell., Vol.11, pp.799-813, 1989.

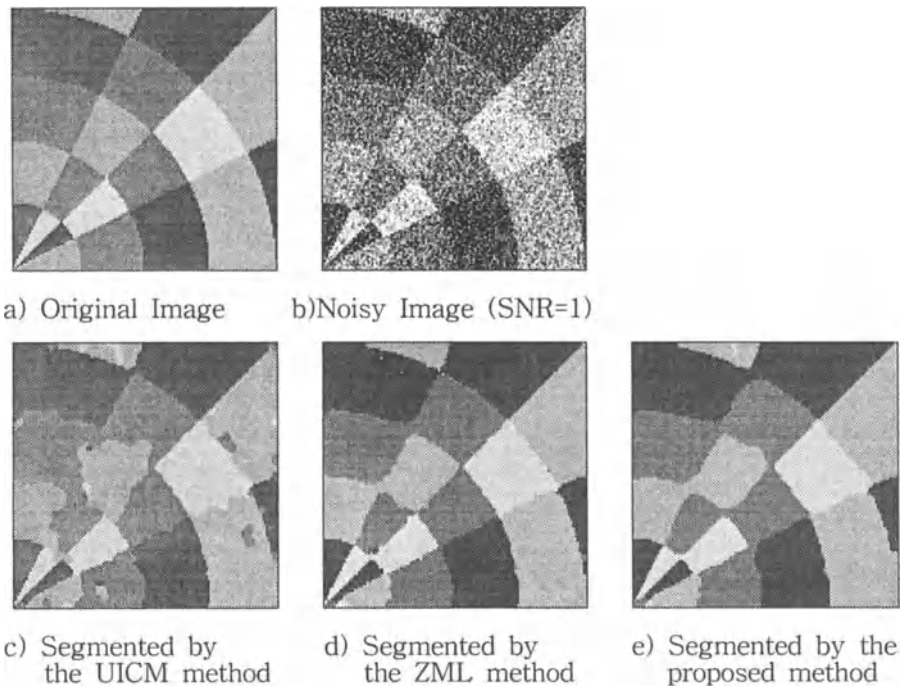


Fig. 3. 4-region noisy image segmentations

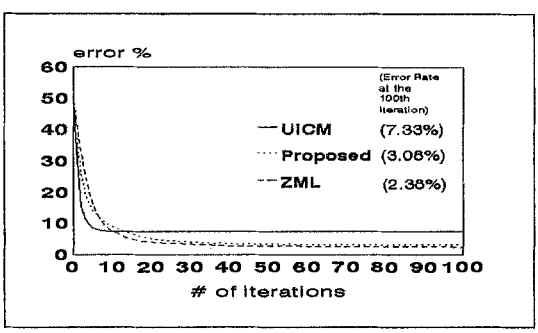


Fig. 4. Comparison of error rates for Fig.3.

Lane Boundary Detection Using Deformable Templates: Effects of Image Subsampling on Detected Lane Edges

Karl Kluge
University of Michigan AI Lab
1101 Beal Avenue
Ann Arbor, MI 48109-2110 (USA)
kckluge@eecs.umich.edu

Sridhar Lakshmanan
Dept. of Electrical and Computer Engineering
University of Michigan -- Dearborn
Dearborn, MI 48128-1491 (USA)
lakshman@umdsun2.umd.umich.edu

Abstract

In order to be robust, a system for detecting road lane boundaries in images needs to be able to handle scenes which contain large amounts of clutter due to shadows, puddles, oil stains, skid marks, leaves and dirt on the road, etc. Many prior systems threshold the image gradient magnitude to detect edges, and then use the detected edge points to identify the lane boundaries. This can result in a very noisy edge image, as there are many situations in which there are strong edges in the image due to irrelevant clutter. There are also many situations in which the edges of the features defining portions of the lane boundaries have a lower intensity gradient than distracting clutter edges in the image. The LOIS (Likelihood of Image Shape) lane detection algorithm solves this problem by using a deformable template approach that uses image intensity gradient information in a way which does not require thresholding. This paper describes the LOIS algorithm in detail, and shows the results of applying the algorithm to a number of challenging scenes. Images are included comparing the lane boundaries detected by applying LOIS to a 240 x 256 image with the lane boundaries detected by applying the algorithm to a 30 x 32 subsampled version of the same image. Qualitatively the results are very similar, but the algorithm runs 45 times faster on the subsampled images.

1 Introduction

Vision-based location of lane boundaries can be divided into two tasks: lane detection and lane tracking. Lane detection is the problem of locating lane boundaries without prior knowledge of the road geometry. Lane tracking is the problem of tracking the lane edges from frame to frame given an existing model of road geometry. Lane tracking is an easier problem than lane detection, as prior knowledge of the road geometry permits lane tracking algorithms to put fairly strong constraints on the likely location and orientation of the lane edges in a new image. Lane detection algorithms, on the other hand, have to locate the lane edges without a strong model of the road geometry, and do so in situations where there may be a great deal of clutter in the image.

Most lane detection algorithms are edge-based (see the discussions of related work in [2] and [3]). In many road scenes it isn't possible to select a threshold which eliminates noise edges without also eliminating many of the lane edge points of interest. In order to overcome this problem, we use a deformable-template approach which eliminates the need to threshold the image gradient magnitudes. This allows weak edge points which are consistent with a prior geometric model of possible lane boundary shapes to overrule strong edge points which are not consistent with the prior geometric model. The deformable-template approach uses a prior model of the set of possible lane shapes in the image plane, a likelihood function which evaluates how

well a possible lane shape matches an image, and an optimization routine which finds the lane shape which maximizes the likelihood function for a given image.

An earlier paper [3] described experiments which were performed to select robust values for the algorithm parameters. This paper focuses on efforts to increase the speed of the algorithm through the use of image subsampling. This improves the speed of the algorithm by decreasing the number of pixels processed.

Section 2 describes the LOIS (*Likelihood of Image Shape*) lane detection system. LOIS uses a deformable template approach which can detect lane boundaries using very simple image processing operations even in cases where there are incomplete lane markings and significant numbers of noise edges.

Section 3 shows representative results of applying LOIS to various road types and environments. The lane boundaries detected by LOIS using a 240 x 256 pixel resolution image are compared to the lane boundaries detected using a 30 x 32 pixel subsampled image.

Section 4 describes directions for future work. Emphasis is placed on implementation on a mobile testbed vehicle, and sensor fusion of millimeter wave radar data with visual data for enhanced reliability in poor visibility conditions.

2 The LOIS lane detection system

The LOIS lane detection algorithm uses a deformable template approach to the lane detection problem in order to handle situations where the lane edges in an image have relatively weak local contrast, or where there are strong distracting edges due to shadows, puddles, pavement cracks, etc. It formulates the lane detection task as the problem of finding the set of lane edge parameters which is most likely given a (possibly very cluttered) image of a road scene. There are three main components to the LOIS algorithm:

- the shape model describing the lane edges, which defines a set of shape parameters;
- the likelihood function, which provides a relative measure of how well a given set of shape parameters match the data in a particular image of a road scene; and
- the optimization algorithm used to determine the best set of lane shape parameters for a given image by maximizing the likelihood function (currently the Metropolis algorithm).

Each of these components is described in detail below. The use of a deformable template model permits the algorithm to achieve good performance using very simple image processing, even in the presence of distracting shadows, texture, pavement cracks, etc.

2.1 The model of lane shape

A common shape model used in lane detection and tracking algorithms assumes that pavement edges and lane markings are defined by concentric circular arcs on a flat ground plane. Under perspective projection these circular arcs on the

ground plane project to curves in the image plane which can be closely approximated by curves of the form

$$column = k'/(row - hz) + b' \times (row - hz) + vp$$

The hz parameter is the row in the image plane corresponding to the horizon of the ground plane. The k' parameter is linearly proportional to the curvature of the arc on the ground plane. The vp parameter is a function of the tangential orientation of the arc on the ground plane, with some coupling to the arc curvature as well (the amount of this coupling depends on the camera tilt). The b' parameter is a function of the offset of the arc from the camera on the ground plane, with couplings to arc curvature and tangential orientation (again, the relative contributions of these couplings depends on the camera tilt) [2]. While the radius of curvature and tangent orientation of the left and right lane edges will differ slightly, constraining the left and right lane edges to have the same k' and vp parameters closely approximates the actual lane edge shapes for all but very small radii of curvature.

The coefficients of the curve defining a lane boundary will change as the spatial resolution of the image changes for a given camera field of view. If

$$c = k_1'/(r - hz_1) + b'_1 \times (r - hz_1) + vp_1$$

is the curve defining a lane boundary in an image at a given spatial resolution, then the curve defining the lane boundary in a subsampled version of that image with $1/zf$ as many rows and columns will be

$$c = \frac{k'_1}{zf^2 \times \left(r - \frac{hz_1}{zf}\right)} + b'_1 \times \left(r - \frac{hz_1}{zf}\right) + \frac{vp_1}{zf}$$

2.2 The likelihood model

The initial likelihood function which was chosen has proven to be fairly robust on marked roads. It encodes the knowledge that the edges of the lane should be near intensity gradients whose orientation should be perpendicular to the lane edge. More formally:

- Define the function $f(\alpha, x) = 1/(1 + \alpha x^2)$.
- Let $gm(r, c)$ be the gradient magnitude at pixel (r, c), with $gd(r, c)$ being the corresponding gradient direction. In the current implementation of LOIS, the gradient magnitude and orientation at each pixel are computed using the Prewitt operator [4]. Other methods, such as the Canny edge detector, could be used if desired.
- Define the edges of the lane in the image by the curves

$$E_L(s) = s_1/(row - hz) + s_2 \times (row - hz) + s_3$$

and

$$E_R(s) = s_1/(row - hz) + s_4 \times (row - hz) + s_3$$

where hz is the row in the image plane corresponding to the horizon under a flat earth assumption.

- Define the prior likelihood function $P(s) = \frac{2}{\pi} \times \operatorname{atan}(\alpha_{width} \times (s_4 - s_2))$.

  This embodies two prior constraints on the lane geometry. The first constraint is the trivial one that the left edge of the lane is to the left of the right edge of the lane. This constraint is enforced by the negative weighting the prior function gives shape parameters which violate it. The second constraint is that the lane is unlikely to be very narrow. This constraint is enforced by the increasing weight given to wider lanes, which asymptotically approaches 1.0 at a rate which depends on α_{width}. The results illustrated in this paper all use $\alpha_{width} = 10.0$.

Given these definitions, the likelihood function for a given image and a given parameter set s is defined as

$$L(s) = P(s) \times \Sigma_{r,c}(gm(r,c) \times f(\alpha_{prox}, c - E_L(s)) \times$$

$$f(\alpha_{orient}, \cos\left(gd(r,c) - \operatorname{atan}\left(\frac{d}{dr}E_L(s)\right)\right)) +$$

$$gm(r,c) \times f(\alpha_{prox}, c - E_R(s)) \times f(\alpha_{orient}, \cos\left(gd(r,c) - \operatorname{atan}\left(\frac{d}{dr}E_R(s)\right)\right))\Big)$$

In other words, the contribution made by a pixel to the likelihood is the gradient magnitude at that pixel, multiplied by a function whose value decreases as the pixel column gets further from the lane edge and a function whose value decreases as the gradient direction at the pixel moves farther away from perpendicular to the tangent to the lane edge. The parameters defining the rate of likelihood fall-off with distance from the lane edges (α_{prox}) and with orientation error (α_{orient}) are set at 0.01 and 1.13 respectively.

The value of α_{prox} depends on the spatial resolution of the image. The value of 0.01 is used for images with 240 rows and 256 columns. When the input image is subsampled to lower resolutions the value of α_{prox} is adjusted in order to produce the same decrease in pixel weight as a function of the distance of a pixel from the expected lane boundary, where the distance is expressed as a fraction of the image width.

While the likelihood function tends to be fairly smooth and convex, there can still be local maxima. As a result, a simple hill-climbing approach to maximizing the likelihood function is inappropriate. The current implementation of LOIS uses the Metropolis algorithm to perform this optimization.

2.3 Optimizing the likelihood using the Metropolis algorithm

The Metropolis algorithm is a stochastic optimization procedure. LOIS uses the Metropolis algorithm with a geometric annealing schedule [5], which operates as follows:

- Let $L(s_N)$ be the likelihood value corresponding to the shape parameters s_N in iteration N of the algorithm;
- In iteration $N+1$:
  - Pick a random point in the shape parameter space in the neighborhood of s_N, $s_N + \Delta$
  - Calculate the "temperature" for iteration $N+1$,

$$T(N+1) = T_{init}\left(\frac{T_{final}}{T_{init}}\right)^{((N+1)/maxiter)}$$

  - Evaluate $\rho(N+1) = e^{\left(\frac{L(s_N+\Delta)-L(s_N)}{T(N+1)}\right)}$
  - Select s_{N+1} according to the following rule:
    - $s_{N+1} = s_N + \Delta$ if $\rho(N+1) \geq 1$
    - $s_{N+1} = s_N + \Delta$ with probability $\rho(N+1)$ if $\rho(N+1) \leq 1$
    - $s_{N+1} = s_N$ otherwise

All the results shown in this paper were generated by running the Metropolis algorithm for 40 iterations. In each iteration a small step away from the current value is tested for each of the lane shape parameters. The initial temperature is 10.0, and the final temperature is 0.01.

Figure 1 illustrates the process of convergence of the Metropolis algorithm on an image with complex shadows. The initial shape parameters specify a somewhat wide straight lane centered in the image. As the iterations of the algorithm improve the value of the likelihood function the shape parameters deform to match the lane markings in the image. It is able to do this despite the presence of shadow edges with much larger contrast than the edges of the double yellow stripe defining the left edge of the lane.

3 Experimental results

Prior experimental work evaluating LOIS [3] focused on systematically testing the effects of varying pairs of algorithm parameters in order to identify what appears to be a robust set of values. These tests used a set of 30 images which represented challenging situations. The results of those tests were used to select the parameters given in this paper.

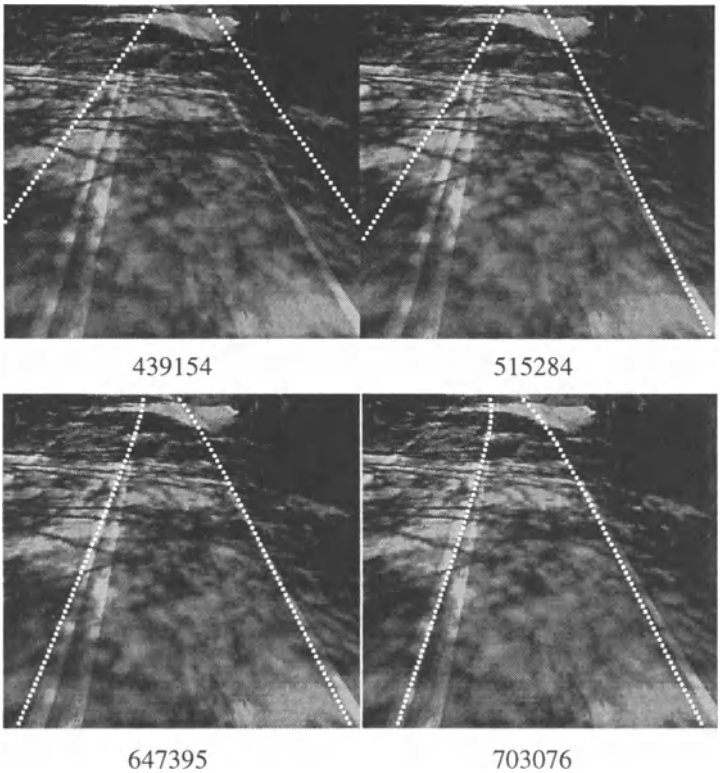


Figure 1: Selected Metropolis iterations showing convergence to the lane edges (likelihood value shown below each image).

Current experiments have focused on examining the impact of using subsampled image data on the recovered lane boundary locations. The table below shows the increase in algorithm speed which can be achieved through the use of subsampled input images. Times were measured on a Sparc 20 (SPEC floating point rating of 78.3).

Resolution (pixels)	Prewitt calc. (seconds)	Metropolis calc. (seconds)
240 x 256	0.57	14.59
120 x 128	0.14	4.04
60 x 64	0.04	1.22
30 x 32	0.01	0.47

Figure 2, Figure 3, and Figure 4 illustrate the results produced by LOIS on 10 images of roads with complex shadows, curvature, broken lane markings, unmarked pavement, and wet pavement with broken markings. The detected lane edges are shown using small white squares spaced along the estimated edge locations. The left image in each pair shows the result produced using a 240 x 256 pixel

resolution input image. The right image in each pair shows the same scene subsampled down to a 30 x 32 pixel resolution. While slight shifts in estimated lane edge location can be seem in several of the images, the results are very close to those produced using the full resolution images while only requiring 1/45th the time to compute.

4 Future work

Having identified what appear to be a fairly robust set of parameters, our current major priority is testing LOIS on a substantially larger data set. We will be using a modified HMMWV as a platform for data collection and testing (Figure 5). The HMMWV has four forward-looking cameras to provide a wide total field of view, and a SPARC 10 processor on board for running image processing algorithms. While the Metropolis algorithm generally performs well as a method of finding an optimal or close to optimal value of the likelihood function, we plan to explore alternative methods of finding the best shape parameters. In particular, we plan to explore the use of branch-and-bound algorithms for optimization.

The deformable template approach used in LOIS is easy to extend to fuse information from multiple sensor modalities. This can be done by formulating a likelihood model which uses registered data from the different sensors to evaluate the relative fit of a given set of shape parameters. We intend to extend the work described here and in [1] to create CLARK (*Combined Likelihood Adding Radar Knowledge*), a system which uses both black and white images and millimeter wave radar data to locate lane and road boundaries. Millimeter wave radar is able to penetrate fog, rain, and dust due to the long wavelength used. It is our expectation that fusing visual and millimeter wave data will produce a system which is more robust in the presence of poor visibility than either sensor modality by itsel

5 Conclusion

Use of a deformable template approach allows LOIS to perform lane detection using intensity gradient information, but without the need to set a threshold and classify pixels as edge or non-edge. A simple likelihood function permits the algorithm to locate lane edges even when contrast is poor or there are many noise edges. A set of algorithm parameters has been found which appears to work in a variety of road environments. The use of a subsampled input image permits faster execution of the algorithms without seriously degrading the quality of the recovered lane edges. The deformable template formulation easily extends to multisensor systems, and work is in progress to add millimeter wave radar data to the intensity data already used in LOIS.

6 Acknowledgments

This research is supported by the National Automotive Center under contract DAAE07-94-C-R064, monitored by TACOM. The authors wish to acknowledge the help of Ms. Geetha Panchapakesan in performing some of the programming and experimental work on the project.

7 References

[1] Grimmer, David; and Lakshmanan, Sridhar. "Finding Straight Edges in Radar Images Using Deformable Templates." In *Proceedings of the 27th Conference on Information Sciences and Systems*, 1993.

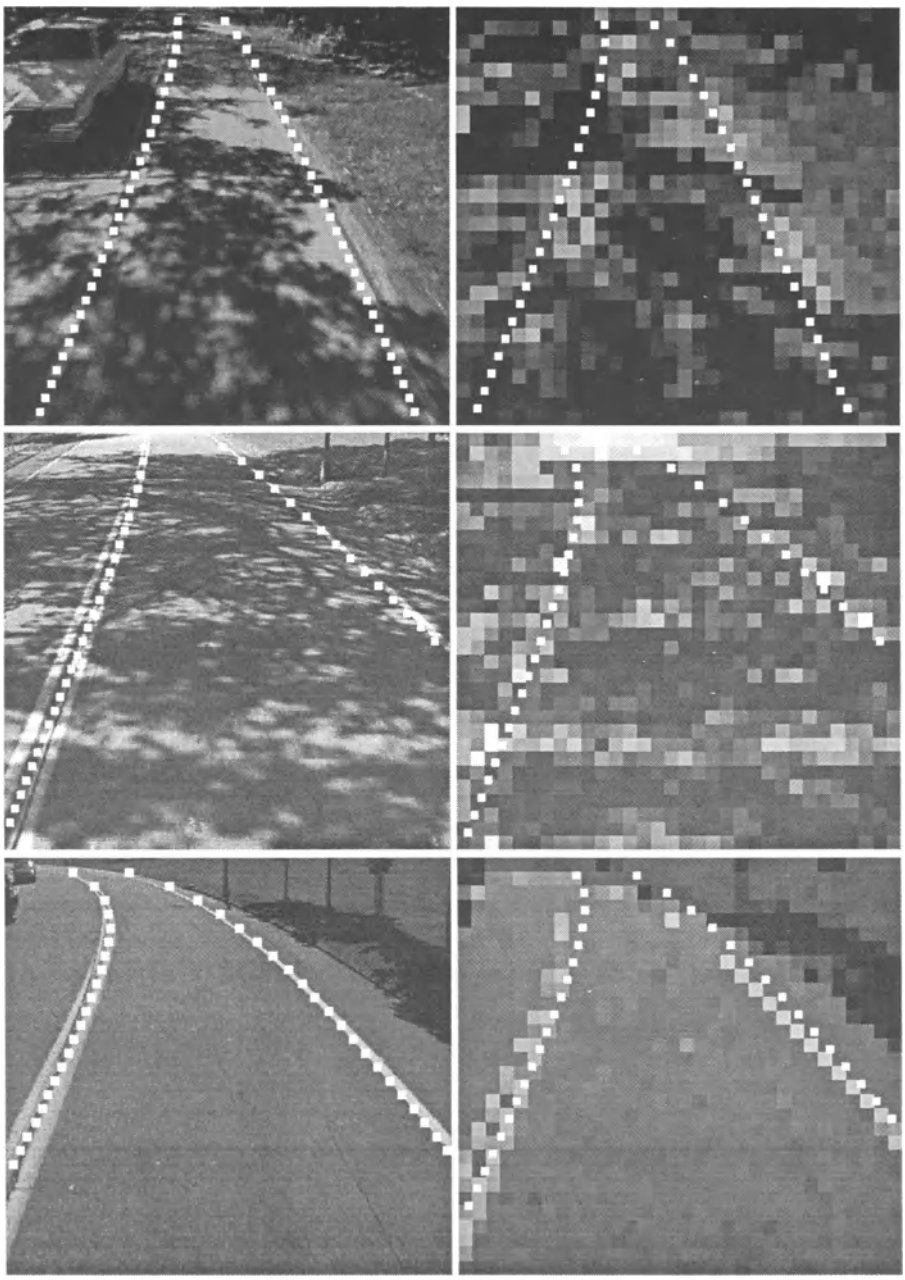


Figure 2: LOIS results, complex shadows (top two image pairs) and road curvature (bottom image pair). The left column shows the detected lane boundaries using a 240 x 256 pixel input image, the right column shows the detected lane boundaries using a 30 x 32 subsampled image of the same scene.

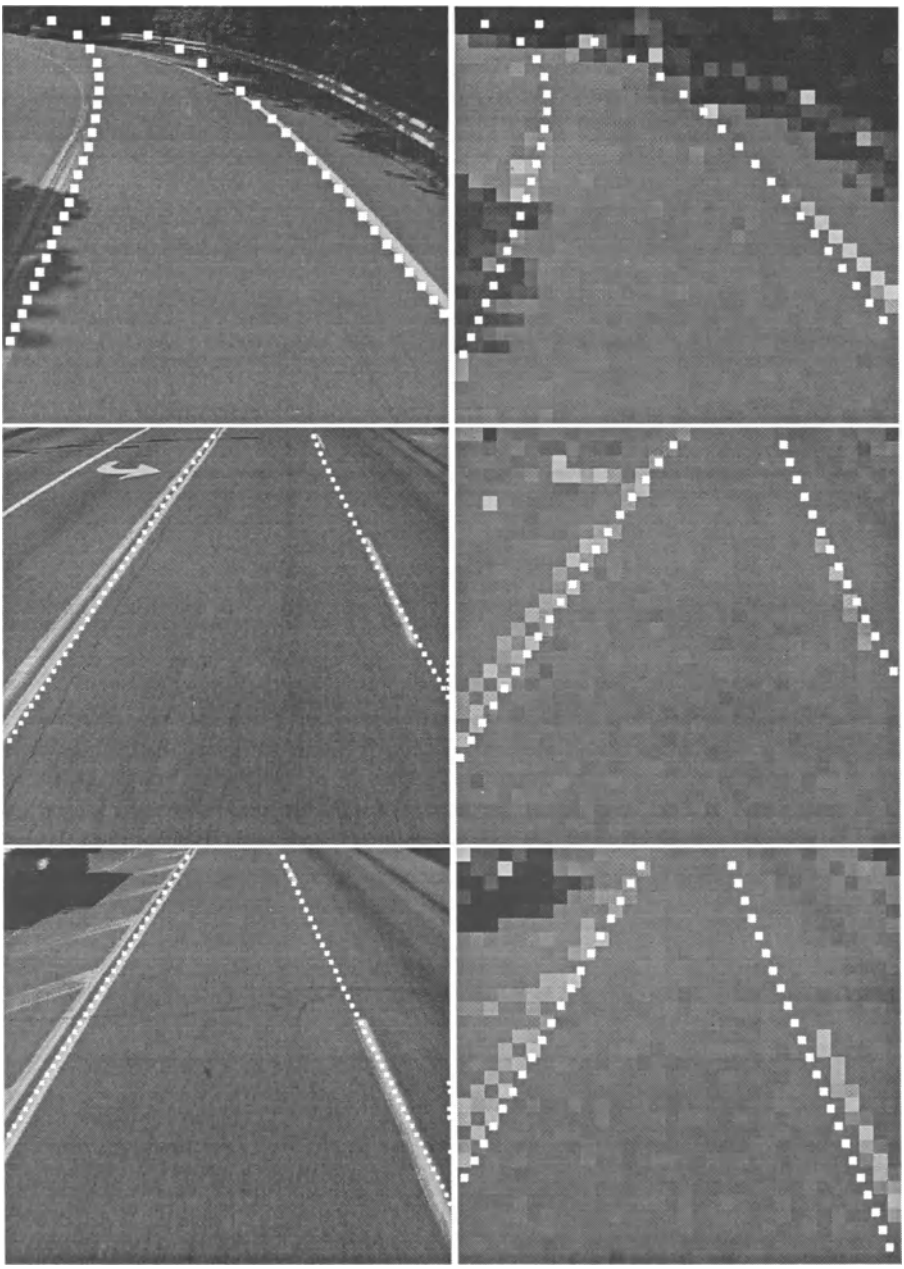


Figure 3: LOIS results, road curvature (top image pair) and broken lane markings (bottom two image pairs). The left column shows the detected lane boundaries using a 240 x 256 pixel input image, the right column shows the detected lane boundaries using a 30 x 32 subsampled image of the same scene.

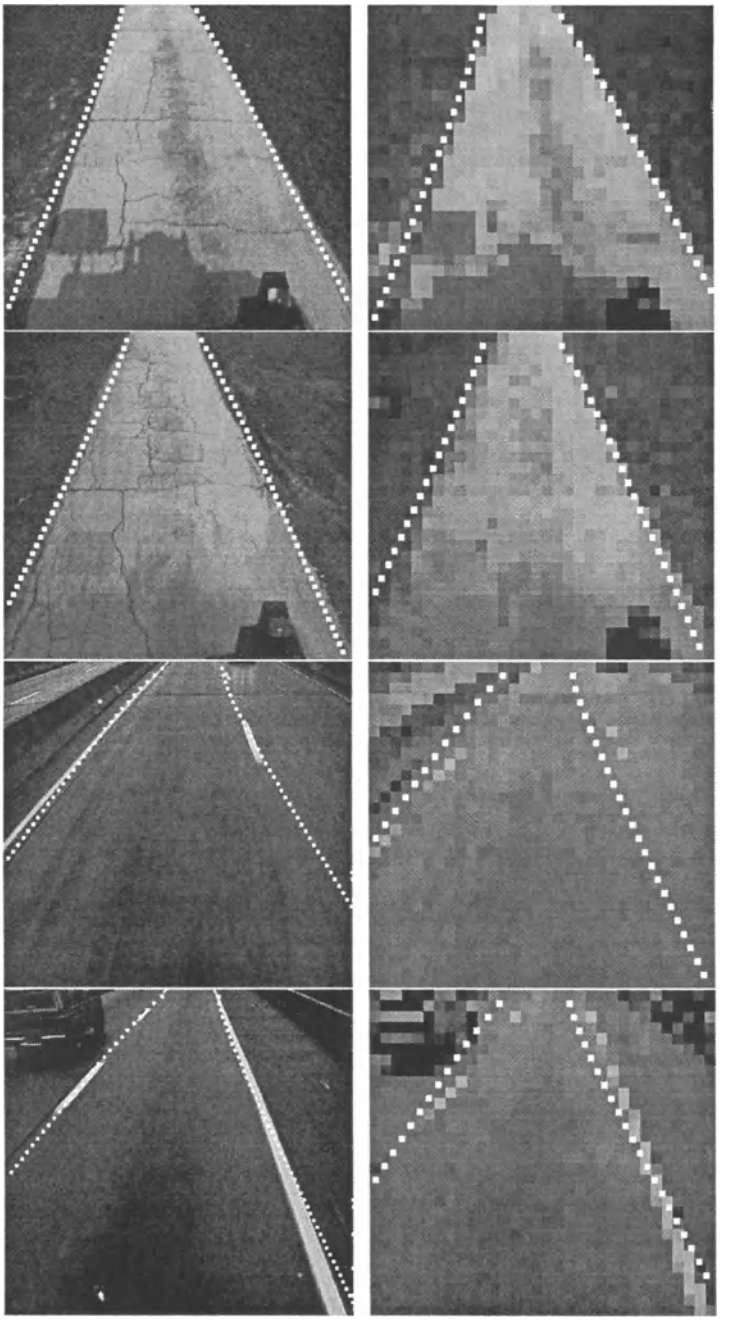


Figure 4: LOIS results, unmarked asphalt bike path (upper two image pairs) and highway with wet pavement, broken lane markings (bottom two image pairs). 240 x 256 pixel input in left column, 30 x 32 pixel input in right column

[2] Kluge, Karl. "Extracting Road Curvature and Orientation From Image Edge Points Without Perceptual Grouping Into Features." In *Proceedings of the Intelligent Vehicles '94 Symposium*, pages 109-114, October 1994.

[3] Kluge, Karl; and Lakshmanan, Sridhar. "A Deformable-Template Approach to Lane Detection." In *Proceedings of the Intelligent Vehicles '95 Symposium*, pages 54-59, September 1995.

[4] Prewitt, Judith M. S. "Object Enhancement and Extraction." in *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, eds., Academic Press, 1970.

[5] Strenski, P. N.; and Kirkpatrick, S. "Analysis of Finite-Length Annealing Schedules." *Algorithmica* 6:346-366, 1991.

Figure 5: HMMWV testbed vehicle.

Contextual Pattern Interpretation

Clique Finding Relaxation Labeling Networks

Marcello Pelillo

Dipartimento di Matematica Applicata e Informatica
Università "Ca' Foscari" di Venezia
Via Torino 155, 30173 Venezia Mestre, Italy

Abstract

The maximum clique problem is a well-known difficult optimization problem which is frequently encountered in computer vision and pattern recognition. In this paper, we show how to approximately solve it by means of a simple relaxation labeling network, of the type originally introduced by Rosenfeld, Hummel, and Zucker. The approach is based on a remarkable result of Motzkin and Straus which allows us to formulate the problem in terms of a certain continuous linearly constrained quadratic optimization problem. Extensive simulations are presented which practically demonstrate the validity of the proposed model.

1 Introduction

Many problems in computer vision and pattern recognition can be formulated in terms of finding a completely connected subgraph (i.e., a *clique*) of a given graph, having largest cardinality. This is called the maximum clique problem (MCP) and is known to be *NP*-hard [1, 2], even to approximate well [3]. One popular approach to object recognition, for example, involves matching an input scene against a stored model, each being abstracted in terms of a relational structure [4, 5], and this problem, in turn, can be conveniently transformed into the equivalent problem of finding a maximum clique of the corresponding *association graph.* This idea was pioneered by Ambler *et al.* [6], and was later developed by Bolles and Cain [7] as part of their local-feature-focus method. Now, it has become a standard technique in computer vision, and has been employing in such diverse applications as stereo correspondence [8], point pattern matching [9], image sequence analysis [10], to name just a few. Another interesting application of the MCP arises in the context of cluster analysis, where graph theoretical methods have long proven to be especially effective [11, 12].

During the past few years, a variety of algorithms have been developed to solve the MCP—Pardalos and Xue [13] have recently offered an excellent and

up-to-date review, with 260 references. Most of these algorithms, however, are sequential in nature and therefore difficult to implement in parallel hardware. We note that there is now increasing interest in developing massively parallel algorithms for approximately solving the MCP, based on neural network models (see [14] and references therein).

In the field of computational vision, relaxation techniques have long been recognized as a viable way of solving various practical problems such as, for example, optical flow computation, edge detection, and stereo vision (see, e.g., [15]). More recently, relaxation models of computation have also attracted considerable interest within the neural network community not only because they appear to be closely related to the style of computation the human brain is supposed to implement [16], but also for their intrinsic abilities to solve difficult optimization problems [17]. One of the earliest and perhaps best-known example of such cooperative algorithms was developed in the mid-1970s by Rosenfeld, Hummel, and Zucker [18] to solve certain labeling problems arising in the 3-D interpretation of ambiguous line drawings. Their *relaxation labeling* model (or variations thereof) has since then been the subject of extensive studies and has now become a widely accepted tool in the computer vision toolkit [19]. Despite the fact that the algorithm was originally motivated by simple heuristic arguments, recent work has shown that it possesses in fact interesting dynamical properties [20].

In this paper, it is seen how a simple yet powerful version of the standard relaxation labeling model of Rosenfeld, Hummel, and Zucker [18] is capable of efficiently finding large cliques in a graph. Central to our approach is a remarkable result due to Motzkin and Straus [21] which allows us to formulate the MCP in terms of a certain continuous optimization problem with linear constraints, a problem that is then naturally mapped onto a relaxation labeling network. This result has also been recently exploited by Pardalos and Phillips [22] to develop a global optimization approach to solve the MCP and derive bounds on the maximum clique size. The proposed approach is attractive not only because of its operational simplicity, but also for its intrinsic locality and parallelism, features that make the model especially suited for parallel hardware implementation. Moreover, in contrast with alternative neural-style clique finding models [14] it has the remarkable feature of being completely devoid of free parameters which are typically difficult to tune properly. Experimental results are presented in the paper which confirm the effectiveness of the model.

2 A simple relaxation labeling network

The relaxation network we shall consider in this paper consists of n pairwise interconnected computational units denoted by $u_1, u_2, \cdots, u_n$. Let w_{ij} be the strength of the connection from unit u_i to unit u_j, and let W denote the $n \times n$ real-valued matrix having w_{ij} as its (i, j) entry. It will be assumed here that the weight matrix is symmetric ($w_{ij} = w_{ji}$) and nonnegative ($w_{ij} \geq 0$). As to the latter point, we note that unipolar networks, like the one we are proposing here,

turn out to be advantageous in many applications, especially when hardware implementation is a concern [23].

Let $u_i(t)$ represent the state of unit u_i at time t, and define the global state of the network at time t to be the n-vector $\mathbf{u}(t) = (u_1(t), \cdots, u_n(t))^T$, where "$T$" denotes transposition. The system operates as follows. It starts out with an initial state vector $\mathbf{u}(0)$ and iteratively and synchronously updates it according to the following discrete dynamical equation

$$u_i(t+1) = \frac{u_i(t)q_i(t)}{\sum_{j=1}^{n} u_j(t)q_j(t)} \qquad (i = 1 \ldots n) \tag{1}$$

where

$$q_i(t) = \sum_{j=1}^{n} w_{ij}u_j(t) \qquad (i = 1 \ldots n) \tag{2}$$

is the net input to unit u_i at time t. The process evolves until $\mathbf{u}(t+1) = \mathbf{u}(t)$, which means that a fixed point has been reached. The reader familiar with relaxation labeling algorithms will soon recognize that the preceding formulas are but a special instance of the standard Rosenfeld-Hummel-Zucker relaxation scheme [18]. In the relaxation labeling terminology, it corresponds to the 1-object n-label case, the role of the compatibility coefficients being played by the nonnegative weights w_{ij}'s. It may also be of some interest to note that the same recurrence relations were independently employed as a model of evolution in population genetics (see, e.g., [24]).

Now, let $\mathcal{K}$ denote the following simplex of $\Re^n$:

$$\mathcal{K} = \left\{ \mathbf{u} \in \Re^n \ : \ u_i \geq 0 \ \forall i = 1 \ldots n, \text{ and } \sum_{i=1}^{n} u_i = 1 \right\} .$$

Because of the normalization factor present in equation (1) and the unipolarity condition, the network performs essentially a mapping of the domain $\mathcal{K}$ onto itself, provided that $\mathbf{u}(0) \in \mathcal{K}$. Levinson, Rabiner, and Sondhi [25], in a rather different context, offered a simple geometrical interpretation for transformations like (1). Let $\mathbf{u}$ be a point in $\mathcal{K}$, and let $\mathbf{q}$ denote the n-vector composed of the q_i's, as defined in (2). Moreover, let $\mathbf{z}$ be the n-vector whose ith component is given by the component-wise product between $\mathbf{u}$ and $\mathbf{q}$, i.e., $z_i = u_i q_i$. Then, it is readily seen that the vector obtained by applying the transformation (1) to $\mathbf{u}$ is simply the intersection of the vector $\mathbf{z}$ (or its extension) with the hyperplane defined by $\sum_i u_i - 1 = 0$.

A direct application of what is known as the Baum-Eagon inequality [26] reveals that the mapping defined by formulas (1) and (2) acts as a *growth transformation* [20] for the following homogeneous quadratic polynomial

$$P(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T W \mathbf{u} \ , \tag{3}$$

which means that

$$P(\mathbf{u}(t+1)) > P(\mathbf{u}(t)) \tag{4}$$

for all $t \geq 0$, unless $\mathbf{u}(t+1) = \mathbf{u}(t)$. This property follows immediately from the fact that, when $W = W^T$, we get

$$\frac{\partial P(\mathbf{u})}{\partial u_i} = \sum_{j=1}^{n} w_{ij} u_j \qquad (i = 1 \ldots n)$$

which have exactly the same form as the q_i's in (2). Put another way, we can say that $-P$ is a Liapunov (or "energy") function for the dynamical system described by equations (1) and (2). As an aside, we mention that the Baum-Eagon inequality has also recently motivated the development of a novel relaxation labeling algorithm [27]. For the sake of completeness, however, we note that this result can also be readily obtained from an earlier (and less general) theorem of Blakley [28].

3 Finding cliques via relaxation

Let an undirected graph $G = (V, E)$ be given, where $V = \{1, \cdots, n\}$ is the set of vertices, and $E \subseteq V \times V$ is the set of edges. Formally, the MCP consists of finding the largest subset $C \subseteq V$ such that for all distinct $i, j \in C$, we have $(i, j) \in E$. Motzkin and Straus [21] proved the following interesting result which relates the MCP to a certain constrained maximization problem: this turns out to be the basis of the work reported in this paper (see also [22] for proof).

Theorem 1 (Motzkin-Straus) *Let k be the size of the maximum clique C in the graph $G = (V, E)$, and consider the following quadratic function*

$$f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T A \mathbf{u} \tag{5}$$

where $A = (a_{ij})$ is the adjacency matrix of G (i.e., $a_{ij} = 1$ if $(i, j) \in E$, and $a_{ij} = 0$ if $(i, j) \notin E$). If α denotes the maximum of $f(\mathbf{u})$ over the simplex $\mathcal{K}$, then we have

$$k = \frac{1}{1 - 2\alpha} \ . \tag{6}$$

Furthermore, this maximum can be attained by setting $u_i = 1/k$ if $i \in C$, and $u_i = 0$ if $i \notin C$.

Hence, the Motzkin-Straus result provides us with a method to solve the "evaluation" version of the MCP, i.e., given a graph determine the *size* of its maximum clique (not the clique itself): this can be accomplished by maximizing in $\mathcal{K}$ the polynomial f, and then using equation (6). We note that this problem formulation is essentially equivalent to its original "optimization" counterpart.

Straightforward recursive or iterative procedures are in fact available to determine the vertices comprising the maximum clique once we have a means to compute its size [2, 22]. More interestingly, the next proposition asserts that if the maximum $\mathbf{u}^*$ of f over $\mathcal{K}$ has a particular form, then we need not resort to such procedures to derive the maximum clique, being it directly encoded into $\mathbf{u}^*$ itself.

Proposition 1 *Let the maximum α of $f(\mathbf{u})$ over $\mathcal{K}$ be attained at a point $\mathbf{u}^*$ of the form $u_i^* = 1/k$ if $i \in C$ and $u_i^* = 0$ if $i \notin C$, for some subset $C = \{c_1, \cdots, c_k\} \subseteq V$. Then C is a maximum clique of G.*

Proof: First, observe that from the structure of $\mathbf{u}^*$ we get

$$\alpha = f(\mathbf{u}^*) \leq \frac{1}{k^2}\binom{k}{2} = \frac{1}{2}\left(1 - \frac{1}{k}\right) ,$$

with equality if and only if C is complete. It is therefore readily seen that no clique of size greater than k can be found in G (this would in fact contradict the hypothesis that $\mathbf{u}^*$ is a maximum of f over $\mathcal{K}$). Consequently, by applying the Motzkin-Straus theorem, we can say that the maximum α is equal to $(1-1/k)/2$ and this implies that C is complete, thereby proving that C is a maximum clique of G.

Note that, in general, global maxima of the Motzkin-Straus problem do not necessarily correspond to maximum cliques. This was earlier observed by Pardalos and Phillips [22], and has recently been formalized by Pelillo and Jagota [29] who showed that the Motzkin-Straus problem can have in fact infinitely many "infeasible" (or spurious) solutions, that is maxima that are not in the form specified in Proposition 1.

Now, once that the MCP is formulated in terms of maximizing a quadratic polynomial over $\mathcal{K}$, the use of the previously defined relaxation model immediately suggests itself. Consider, in fact, a network with as many units as there are vertices in the graph G, and define the (symmetric and nonnegative) weight matrix to be

$$W = A ,$$

where A is the adjacency matrix of G. From the preceding discussion we know that the network, starting from an arbitrary initial state $\mathbf{u}(0)$, will iteratively maximize the function $f(\mathbf{u})$ defined in (5), and will eventually approach a fixed point $\mathbf{u}^*$ corresponding to a maximum of f—note that we are not in principle guaranteed that this will also correspond to the globally optimal solution. This solution can then be employed to get an estimate of the maximum clique size, as seen in Theorem 1. Additionally, if the converged solution $\mathbf{u}^*$ has the form specified in Proposition 1, we are even able to directly determine the clique itself.

Observe that, since the process cannot leave the boundary of $\mathcal{K}$, it is preferable that the relaxation process begins with an interior point (i.e., $u_i(0) > 0$ for

all i). A reasonable choice is to make the process start out in the vicinity of the center of the simplex $\mathcal{K}$: this would avoid that the search be biased in favor of any particular vertex in G.

4 Results

In order to assess the effectiveness of the proposed clique finding model, extensive simulations were carried out over randomly generated graphs of various sizes and densities.[1] The results were compared with those obtained using well-known exact clique finding algorithms; in particular, the Bron-Kerbosch (BK) backtracking-based procedure [30], and the partially enumerative algorithm developed by Carraghan and Pardalos (CP) [31] were employed for comparison. We found the BK algorithm attractive because it is able to generate *all* cliques of a graph, and this allowed us to better understand certain phenomena associated with our relaxation model. When the computational time of BK became prohibitively long, we resorted to the CP algorithm, which is considerably faster than BK, but finds just *one* maximum clique.

The sizes considered in our experiments were $n = 10, 25, 50, 75, 100, 250, 500$ and the densities were $\delta = 0.10, 0.25, 0.50, 0.75, 0.90$. Indeed, for 250- and 500-vertex graphs the densities 0.75 and 0.90 were excluded, because even the CP algorithm became excessively slow in these cases. For each pair (n, δ), 100 different graphs with n vertices and approximate density δ were obtained so that, overall, 3100 graphs were employed. For each of them, the corresponding relaxation network was constructed as described in the previous section and then run by starting the process from the vector

$$\mathbf{u}(0) = (1/n, \cdots, 1/n)^T ,$$

which corresponds to the center of the domain $\mathcal{K}$. We tried also to make the relaxation search begin with noisy versions of this vector, but no significant difference in performance was observed. To get a high-precision final solution, the network was allowed to iterate until the (squared) distance between two successive states became smaller than 10^{-10}. Each of the 3100 graphs was also given as input to the BK/CP algorithms in order to determine its actual maximum clique size k and then compute the corresponding global maximum α, according to the Motzkin-Straus theorem. Specifically, the BK algorithm was run over all graphs whose size n was less than or equal to 100; the exception was the 100-vertex 0.90-density case where, for reasons of computational expediency, we resorted to the CP algorithm. The CP algorithm was also employed for all 250- and 500-vertex graphs.

To systematically evaluate the goodness of the solutions found by the network, the following recently proposed "quality" measure was adopted [32]:

[1] The density δ of a graph G with n vertices is defined to be the ratio between the number of edges in G and $\binom{n}{2}$.

Table I

Average values of the Q-measure for various sizes and densities

n \ δ	0.10	0.25	0.50	0.75	0.90
10	0.99	0.99	1.00	1.00	1.00
25	0.99	0.99	0.99	1.00	1.00
50	0.99	0.99	0.99	1.00	1.00
75	0.97	1.00	0.96	1.00	0.99
100	0.92	0.97	0.90	1.00	1.00
250	0.93	0.92	0.81	—	—
500	0.92	0.92	0.80	—	—

$$Q = \frac{f_{ave} - f_{relax}}{f_{ave} - \alpha},$$

where f_{ave} is the average value of the function f calculated by collecting a sufficient number of random points in $\mathcal{K}$, f_{relax} is the value of f associated with the solution found by the relaxation network, and α (as usual) represents the global maximum of f over $\mathcal{K}$ calculated using the BK/CP algorithms. From the above definition, it is apparent that values of Q near to zero correspond to very poor solutions (i.e., $f_{ave} \approx f_{relax}$), whereas values of Q near to one correspond to very good solutions (i.e., $f_{relax} \approx \alpha$).

The results of the simulations are summarized in Table I. Each row in the table represents a particular value of n, each column represents a particular value of δ, and the entry corresponding to the pair (n, δ) contains the mean value of Q, averaged over the 100 trials. As can be seen, the relaxation labeling network, despite its inability to escape from local optima, did a good job of determining the size of the maximum cliques, even for high values of n and δ. This suggests that global maxima of the objective function f have large basins of attraction. In Table II, the median number of iterations performed by the network to get a solution, for all cases considered, is reported. Note that the stopping criterion used in this study was a rather "strict" one; in fact, the number of relaxation iterations can be reduced considerably by increasing the stopping threshold (which in this case was fixed at 10^{-10}) at the expenses of getting less accurate final solutions.

As we have noted in the previous section, there may be a number of spurious solutions associated with the Motzkin-Straus program [29] and, in principle, nothing prevents the relaxation process at converging towards one such solution. It should be clear that spurious solutions represent an undesirable feature as they do not permit us to recover the vertices comprising a clique directly from their structure. The use of the BK algorithm over graphs with at most 100 vertices allowed us to discover certain interesting regularities in the behavior of our algorithm. Specifically, we observed that when a graph contained either a

Table II
Median number of relaxation iterations to get a solution

n \ δ	0.10	0.25	0.50	0.75	0.90
10	54	54	56	99	119
25	36	64	118	175	224
50	53	84	153	268	367
75	59	98	160	284	513
100	82	112	187	369	552
250	96	150	223	—	—
500	102	160	262	—	—

unique maximum clique or two or more disjoint maximum cliques, the network systematically converged to a fixed point $\mathbf{u}^*$ having the form specified in Proposition 1. This made it possible to determine the nodes of one such clique without further processing. Unfortunately, when two or more maximum cliques shared some vertices this did not happen. However, we found that in such cases the network typically approaches a solution where all nonzero components correspond to the nodes of these cliques and, additionally, the components associated with the shared vertices have maximum value (typically of the form $1/k$, k being the clique size). This allows one to determine the maximum clique with little additional effort. We note that these empirical findings are perfectly in accordance with certain theoretical results presented in [29].

5 Conclusions

We have shown how the classical problem of finding the maximum clique of an undirected graph can be solved by means of a simple yet powerful Rosenfeld-Hummel-Zucker relaxation labeling network. This is a parallel distributed computational model that is widely popular within the machine vision domain, and has recently proven to possess interesting dynamical properties. Based on a result due to Motzkin and Straus, we have seen how the MCP can be naturally mapped onto a relaxation labeling network which is capable of maximizing quadratic polynomials over a certain linearly constrained domain. One remarkable feature of the algorithm presented here is that, unlike other parallel clique finding relaxation models [14], it is completely devoid of free parameters which are typically difficult to determine, and this makes it extremely simple to implement and apply. Extensive simulations over randomly generated graphs have practically demonstrated the effectiveness of the proposed approach.

Acknowledgment: The author is grateful to S. Ricci for carrying out the simulations presented in the paper.

References

[1] M. R. Garey and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York.

[2] C. H. Papadimitriou and K. Steiglitz (1982). *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ.

[3] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy (1991). Approximating clique is almost NP-complete. *Proc. 32nd Ann. Symp. Found. Comput. Sci.*, San Juan, Puerto Rico, 2–12.

[4] R. T. Chin and C. R. Dyer (1986). Model-based recognition in robot vision. *Computing Surveys* **18**(1), 67–108.

[5] P. Suetens, P. Fua, and A. J. Hanson (1992). Computational strategies for object recognition. *Computing Surveys* **24**(1), 5–61.

[6] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone (1973). A versatile computer-controlled assembly. *Proc. Third Int. J. Conf. Artif. Intell.*, Stanford, CA, 298–307.

[7] R. C. Bolles and R. A. Cain (1982). Recognizing and locating partially visible objects: The local-feature-focus method. *Int. J. Robotics Res.* **1**(3), 57–82.

[8] R. Horaud and T. Skordas (1989). Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Pattern Anal. Machine Intell.* **11**(11), 1168–1180.

[9] H. Ogawa (1986). Labeled point pattern matching by Delaunay triangulation and maximal cliques. *Pattern Recognition* **19**(1), 35–40.

[10] B. Radig (1984). Image sequence analysis using relational structures. *Pattern Recognition* **17**(1), 161–167.

[11] J. G. Augustson and J. Minker (1970). An analysis of some graph theoretical cluster techniques. *J. ACM* **17**(4), 571–588.

[12] A. K. Jain and R. C. Dubes (1988). *Algorithms for Clustering Data.* Prentice-Hall, Englewood Cliffs, NJ.

[13] P. M. Pardalos and J. Xue (1994). The maximum clique problem. *J. Global Optim.* **4**, 301–328.

[14] A. Jagota (1995). Approximating maximum clique with a Hopfield network. *IEEE Trans. Neural Networks* **6**(3), 724–735.

[15] D. H. Ballard and C. M. Brown (1982). *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ.

[16] T. J. Sejnowski (1986). Open questions about computation in cerebral cortex. In J. McClelland and D. E. Rumelhart (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition— Vol. 2* (pp. 372–389). MIT Press, Cambridge, MA.

[17] J. J. Hopfield and D. W. Tank (1985). "Neural" computation of decisions in optimization problems. *Biol. Cybern.* **52**, 141–152.

[18] A. Rosenfeld, R. A. Hummel, and S. W. Zucker (1976). Scene labeling by relaxation operations. *IEEE Trans. Syst. Man Cybern.* **6**(6), 420–433.

[19] J. Kittler and J. Illingworth (1985). Relaxation labeling algorithms—A review. *Image Vision Comput.* **3**(4), 206–216.

[20] M. Pelillo (1994). Nonlinear relaxation labeling as growth transformation. *Proc. 12th Int. Conf. Pattern Recognition*, Jerusalem, Israel, II:201–206.

[21] T. S. Motzkin and E. G. Straus (1965). Maxima for graphs and a new proof of a theorem of Turán. *Canad. J. Math.* **17**, 533-540.

[22] P. M. Pardalos and A. T. Phillips (1990). A global optimization approach for solving the maximum clique problem. *Int. J. Computer Math.* **33**, 209–216.

[23] J. S. Denker (1986). Neural networks refinements and extensions. In J. S. Denker (Ed.), *Neural Networks for Computing* (pp. 121–128). American Institute of Physics, New York.

[24] W. J. Ewens (1979). *Mathematical Population Genetics.* Springer-Verlag, Berlin.

[25] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.* **62**(4), 1035–1074.

[26] L. E. Baum and J. A. Eagon (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.* **73**, 360–363.

[27] A. J. Stoddart, M. Petrou, and J. Kittler (1995). A new algorithm for probabilistic relaxation based on the Baum Eagon theorem. In V. Hlavác and R. Sára (Eds.), *Computer Analysis of Images and Patterns* (pp. 674–679). Springer, Berlin.

[28] G. R. Blakley (1964). Homogeneous nonnegative symmetric quadratic transformations. *Bull. Amer. Math. Soc.* **70**, 712–715.

[29] M. Pelillo and A. Jagota (1995). Feasible and infeasible maxima in a quadratic program for the maximum clique problem. *J. Artif. Neural Networks*, Special issue on "Neural Networks for Optimization," submitted for publication.

[30] C. Bron and J. Kerbosch (1973). Finding all cliques of an undirected graph. *Commun. ACM* **16**(9), 575–577.

[31] R. Carraghan and P. M. Pardalos (1990). An exact algorithm for the maximum clique problem. *Oper. Res. Lett.* **9**, 375–382.

[32] P. W. Protzel (1990). Comparative performance measure for neural networks solving optimization problems. *Proc. Int. J. Conf. Neural Networks*, Washington, DC, II:523–526.

Rectifying Structural Matching Errors

Edwin R. Hancock and Richard C. Wilson
Department of Computer Science, University of York, Y01 5DD, UK.

Abstract. Structural errors which arise due to poor image segmentation or clutter pose one of the main obstacles to effective relational graph matching. The aim of this paper is to provide a comparative evaluation of a number of contrasting approaches to the control of structural errors. Unique to this study is the way in which we show how a diverse family of algorithms relate to one-another using a common Bayesian framework. According to our adopted Bayesian approach relational consistency is gauged by Hamming distance. We illustrate three different ways in which this consistency measure may be used to rectify structural errors. The main conclusion of our study is that the active process of graph-editing outperforms the alternatives in terms of its ability to effectively control a large population of contaminating clutter.

1 Introduction

Relational structure matching has been a task of pivotal importance since the inception of machine vision. It was the seminal paper of Barrow and Popplestone [1] that first established the relational graph as a practical representation for scene matching. However, it was soon appreciated that structural matching is only effective when exact relational descriptions are to hand. It was precisely this observation that stimulated much of the classical work on structural pattern recognition of the 1980's which aimed to realise relational matching by inexact means. Shapiro and Haralick not only identified the practical difficulties associated in the search for subgraph isomorphisms when the graphs under match are corrupted realisations of one-another [7], but also proposed a relational distance metric which could be used to gauge structural differences [8]. According to Shapiro and Haralick [7], missing or extraneous entities are accommodated by inserting null nodes into the graphs without penalty. Sanfeliu and Fu [6] use a more complex set of structural edit operations to transform corrupted graphs into their exact equivalents. There are individual costs associated with node relabelling, node deletion or insertion together with the analogous operations for edges.

We have recently reported a novel Bayesian framework for structural matching which gauges the cost of symbolic consistency in a much finer way [9, 10, 11]. This new framework for relational matching not only draws on Hamming distance as a measure of relational consistency, it is also robust to the structural errors caused by noise or clutter. There are a number of ways in which this new measure of relational consistency may be exploited to enhance the matching process when structural errors are a limiting factor. The most straightforward of these is to attempt to locate the most consistent matches while simultaneously

labelling clutter in an optimisation process [9]. This is the strategy adopted by Boyer and Kak [3], and, by Christmas, Kittler and Petrou [5]. Here the structure of the graphs remains unmodified in the matching process; in other words, extraneous nodes are simply tagged with a null-label. A second strategy is to use the consistency measure to identify candidate matches for inclusion in an association graph; clutter may then be rejected by employing a constraint filtering process [2, 10] similar to that originally described by Barrow and Popplestone [1]. This is a hybrid technique which combines optimisation ideas with classical constraint filtering techniques. A third alternative, is to use the consistency measure to control the reconfiguration of the graphs in an active matching strategy [11] using graph-edit operations [6]. In this way clutter is excluded by deleting nodes and reconfiguring the edge set of the graphs. Although this process has many of the conceptual ingredients of the graph edit process of Sanfeliu and Fu [6], it differs in two important respects. Firstly, the Bayesian measure of relational consistency implicitly gauges the cost of structural errors using only a single model parameter. Secondly, the Bayesian framework directly couples the structural edit operations to the attributes of the raw image entities. Our aim in this paper is therefore to show these alternative strategies for controlling clutter can be accommodated within our framework for modelling relational consistency. Moreover, we offer some experimental comparison of their relative robustness to noise and clutter.

2 Relational Graphs

We abstract the matching process in terms of attributed relational graphs [3, 5, 8]. We use the triple $G = (V, E, \mathcal{A})$ to denote the graphs under match, where V is the set of nodes, E is the set of edges and $\mathcal{A} = \{\underline{x}_i, \forall i \in V\}$ is a set of unary measurements associated with the nodes. Our aim in matching is to associate nodes in a graph $G_1 = (V_1, E_1, \mathcal{A}_1)$ representing data to be matched against those in a graph $G_2 = (V_2, E_2, \mathcal{A}_2)$ representing an available relational model. This matching process is facilitated using constraints provided by suitable relational subunits of the model graph G_2. Formally, the matching is represented by a function $f : V_1 \rightarrow V_2$ from the nodes in the data graph G_1 to those in the model graph G_2.

In performing the matches of the nodes in the data graph G_1 we will be interested in exploiting structural constraints provided by relational units of the model graph G_2. In selecting appropriate units, we will strike a compromise between fineness of representation, on the one hand, and computational expense, on the other. For this reason we will use subgraphs that consist of neighbourhoods of nodes interconnected by edges; for convenience we refer to these structural subunits or N-ary relations as super-cliques. The super-clique of the node indexed j in the graph G_1 with edge-set E_1 is denoted by the set of nodes $C_j = j \cup \{i|(i,j) \in E_1\}$. We use the notation $R_j = (u_1, u_2, ..., u_{|C_j|})$ to denote the N-ary symbolic relation represented by the nodes of the super-clique $C_j \subset V_1$ in the data graph G_1. The matched realisation of this super-clique is denoted by

the relation $\Gamma_j = (f(u_1), f(u_2), ..., f(u_{|C_j|}))$.

Our aim is to modify the match to optimise a measure of global consistency with the constraints provided by the model graph G_2. The constraints available to us are provided by the N-ary symbol relations on the super-cliques of the model graph G_2. The critical ingredient in developing our matching scheme is the set of feasible mappings between each super-clique of graph G_1 and those of graph G_2. The set of feasible mappings, or dictionary, for the super-clique C_j is denoted by $\Theta_j = \{S_i\}$ where $S_i = i \cup \{j|(i,j) \in E_2\}$. Each element S_i of Θ_j, is therefore a relation formed on the nodes of the model graph; we denote such consistent relations by $S_i = (v_1, v_2, ...)$. The dictionary of feasible mappings for the super-clique C_j consists of all the consistent relations that may be elicited from the graph G_2. In practice these relations are formed by performing cyclic permutation of the non-centre nodes for each super-clique of the requisite size (an example of this mapping process is shown in Figure 1); this process effectively preserves the adjacency structure of planar model graphs while leaving dictionary invariant to potential scene translations, scalings or rotations.

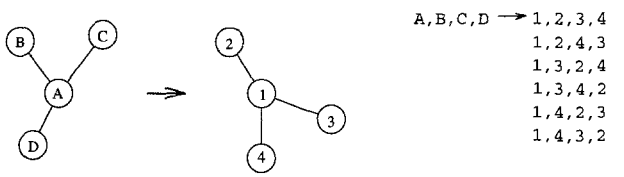


Fig. 1. Example super-clique mapping

3 Relational Consistency

Our modelling of structural consistency is is Bayesian and commences from the joint prior for the super-clique matches [9, 10, 11], i.e. $P(\Gamma_j)$. This model of the matching probability can be viewed as providing a means of imposing constraints on consistent relational matches. Rather than imposing attribute constraints, we draw on an objective Bayesian model of relational corruption which is posed at the symbolic level. As we noted in Section 2, the consistent labellings available for gauging the quality of match are represented by the set of relational mappings from C_j onto G_2, i.e. Θ_j. As demanded by the Bayes rule, we compute the probability of the required super-clique matching by expanding over the basis configurations belonging to the dictionary Θ_j

$$P(\Gamma_j) = \sum_{S_i \in \Theta_j} P(\Gamma_j|S_i).P(S_i) \tag{1}$$

The development of a useful graph-mapping measure from this expression requires models of the processes at play in matching and of their roles in producing errors. We assume that the various types of matching error for nodes

belonging to the same super-clique are memoryless. In direct consequence of this assumption, we may factorize the required probability distribution over the symbolic constituents of the relational mapping under consideration. As a result the conditional probabilities $P(\Gamma_j|S_i)$ may be expressed in terms of a product over label confusion probabilities

$$P(\Gamma_j|S_i) = \prod_{k=1}^{|S_i|} P(f(u_k)|v_k) \tag{2}$$

Our next step is to propose a two component model of the processes which give rise to erroneous matches. The first of these processes is initialisation error, which we aim to rectify by iterative label updates. We assume that initialisation errors occur with a uniform and memoryless probability P_e. The second source of error is structural disturbance of the relational graphs caused by noise, clutter or segmentation error. It is the rectification of these errors which poses the greatest challenge to relational matching. It is not only the motivation of the structural edit operations of Sanfeliu and Fu [6], but also of the null match process of Boyer and Kak [3] and of Christmas, Kittler and Petrou [5]. It is the effective control of these structural errors which is our main concern in Section 4. For the time being we assume that structural errors can also be modelled by a uniform distribution which occurs with probability P_ϕ. Under these dual assumptions concerning the nature of matching errors the confusion probabilities appearing under the product of equation (2) may be assigned according to the following distribution rule

$$P(f(u_k)|v_k) = \begin{cases} P_\phi & \text{if } f(u_k) = \phi \\ (1-P_\phi)(1-P_e) & \text{if } f(u_k) = v_k \\ (1-P_\phi)P_e & \text{if } f(u_k) \neq v_k \end{cases} \tag{3}$$

As a natural consequence of this distribution rule the joint conditional probability is a function of two physically meaningful variables. The first of these is the Hamming distance $H(\Gamma_j, S_i) = \sum_{k \in C_j}(1 - \delta_{f(u_k),v_k})$ which counts the number of conflicts between the current matching assignment Γ_j residing on the super-clique C_j and those assignments demanded by the relational mapping S_i. The second variable is the number of null matches assigned to the relational mapping Γ_j, which we denote by $\Psi(\Gamma_j)$. With these two ingredients, the resulting expression for the joint conditional probability acquires an exponential character

$$P(\Gamma_j) = \frac{K_{C_j}}{|\Theta_j|} \sum_{S_i \in \Theta_j} \exp[-(k_e H(\Gamma_j, S_i) + k_\phi \Psi(\Gamma_j)] \tag{4}$$

where $K_{C_j} = [(1-P_e)(1-P_\phi)]^{|C_j|}$. The two exponential constants appearing in the above expression are related to the matching-error probability and the null match probability, i.e. $k_e = \ln \frac{(1-P_e)}{P_e}$ and $k_\phi = \ln \frac{(1-P_e)(1-P_\phi)}{P_\phi}$. The probability distribution may be regarded as providing a natural way of softening the hard relational constraints operating in the model graph. The most striking and critical feature of the expression for $P(\Gamma_j)$ is that the consistency of match is gauged

by a series of exponentials that are compounded over the dictionary of consistently mapped relations. It is this feature that distinguishes it from alternatives reported elsewhere in the literature [3, 5].

4 Controlling Structural Errors

Our aim in this section is to describe three contrasting strategies which exploit the consistency concept to overcome structural errors during the matching process [9, 10, 11]. The first of these draws directly upon the rather naive uniform probability model described in Section 3 and attempts to label clutter segments with a null label ϕ while simultaneously correcting initialisation errors [9]. The second strategy uses the classical concept of the association graph [2]. Operating without regard to null matches, i.e. with $P_\phi = 0$, we first attempt to correct initialisation errors and then construct the association graph for the optimal set of matches. Clutter is then rejected in a post-processing operation that involves searching for maximal subgraphs of the association graph [10]. The final strategy is to use our consistency measure to identify nodes that are potential clutter tokens. By deleting these nodes from the graph and reconfiguring the edge-set we perform an active matching process [11]. This latter technique aims to simultaneously optimise not only the match of the graph but also to restore its intrinsic relational structure so as to rectify both sources of error described in Section 3.

4.1 Null labelling by optimisation

Our aim here is to simultaneously both rectify initialisation error and tag clutter in an optimisation process. We take as our criterion the *posteriori* probability of match $P(f|\mathcal{A}_1, \mathcal{A}_2)$. Details of the development of the matching scheme have already been described in [9]. Suffice to say that the assignment of nodes to optimise the global MAP criterion may be realised on a node-by-node basis. The match $f(u)$ is assigned as follows

$$f(u) = \arg \max_{v \in V_2 \cup \phi} \frac{P(u, v|\underline{x}_u^1, \underline{x}_v^2)}{P(u, v)} \sum_{j \in C_u} \frac{K_{C_j}}{|\Theta_j|} \sum_{S_i \in \Theta_j} \exp[-(k_e H(\Gamma_j, S_i) + k_\phi \Psi(\Gamma_j)] \tag{5}$$

In the above expression, $P(u, v|\underline{x}_u^1, \underline{x}_v^2)$ are *a posteriori* matching probabilities derived from the unary attributes $\underline{x}_u^1$ for the object u in graph G_1 and $\underline{x}_v^2$ for the object v in graph G_2. The quantity $P(u, v)$, on the other hand, is the *a priori* probability for the match $f(u) = v$. In applying the decision rule of equation (5), the control of the parameters P_e and P_ϕ becomes an issue of critical importance in regulating the number of null matches. Analysis of equation (4), reveals that it does not become energetically favourable for an erroneous match to switch to the null category until $P_e < \frac{P_\phi}{1-P_\phi}$. Since our ultimate aim is eliminate initialisation error, we take the view that the parameter P_e should be treated in the

spirit of a control variable much in the spirit of temperature in an annealing process [4]. Accordingly, we reduce P_e to a small terminal value according to some deterministic iteration dependent schedule. The probability of structural errors P_ϕ, on the other hand, is a fixed property of the graphs and should remain at a constant value that reflects prior expectations concerning the level of structural corruption.

The null-labelling idea is most effective when the data represents a subgraph of the model. Here there is little structural corruption of the structure preserving mappings in the dictionary and the probability distribution appearing in equation (3) is relatively justifiable. In other words, the process can accommodate initialisation errors but not significant structural errors.

4.2 Constraint filtering

The optimisation scheme described in the previous subsection iteratively corrects initialisation errors while identifying clutter nodes as belonging to the null category. One of the difficulties with this scheme lies in controlling the value of the null match probability P_ϕ. If this is set too high then there is a danger of labelling the entirety of nodes as null. If on the other hand the value of P_ϕ is too low then a substantial fraction of clutter nodes will remain mismatched. An alternative strategy which is less demanding in terms of parameter control is to locate a set optimal matches with disregard to the null label ϕ. Clutter is then removed by subsequent a constraint filtering aimed at identifying consistently matched subgraphs. One strategy that suggests itself is to draw on the association graph idea of Barrow and Burstall [2]. Nodes in the association graph represent putative matches; the edges indicate that the matched nodes are connected in both the data and model graphs. Consistently matched subgraphs are maximal cliques of the association graph.

Our strategy in exploiting the association graph to identify clutter is as follows. We first locate a set of matches so as to optimise the configuration probability given in equation (4) by setting the value of P_ϕ to zero, i.e. by disregarding the null-label ϕ. In other words

$$f(u) = \arg\max_{v \in V_2} \frac{P(u, v|\underline{x}_u^1, \underline{x}_v^2)}{P(u, v)} \sum_{j \in C_u} \frac{K_{C_j}}{|\Theta_j|} \sum_{S_i \in \Theta_j} \exp[-k_e H(\Gamma_j, S_i)] \qquad (6)$$

The net anticipated effect of this process is to eliminate initialisation errors. However, residual inconsistencies persist in the form of the erroneous matching of nodes for which no feasible match exists. We therefore require a post-processing step which discards unmatchable elements to a null category. This is applied once the optimisation process reaches convergence and no further iterative improvements in the consistency of match occur. In keeping with the philosophy of the association graph we commence by forming a new graph $G'_1 = (V'_1, E'_1)$ which contains the consistently labelled portions of G_1. To form G'_1 we first eliminate edges whose mapping does not appear in G_2 to form the new edge-set $E_1{}' = \{(u_1, u_2)|(f(u_1), f(u_2)) \in E_2\}$. We then remove disjoint nodes which are

no longer connected by an edge; these nodes have no support and may therefore be considered to be unmatchable. The new node-set consists of the set of nodes $V_1{}' = \{u|(u,v) \in E_1{}', v \in V_1\}$. Disjoint nodes are consigned to the null category, i.e. $u \notin V_1{}' \Rightarrow f(u) = \phi$.

The graph $G_1{}'$ now consists of a number of internally connected yet potentially disjoint patches, in which all the nodes satisfy the edge consistency constraint. Suppose that Λ_i denotes the index-set of one of these disjoint graph partitions. If there are p such partitions, then $V_1' = \bigcup_{i=1}^{p} \Lambda_i$ and there are no interconnecting edges, i.e if $i \neq j$ then $(\Lambda_i \times \Lambda_j) \cap E_1' = \emptyset$. If correct matching were the only process which generated consistency, then $G_1{}'$ would contain only the correctly labelled portion of the graph. However, a small amount of spurious consistency is generated from local matches between unmatchable segments and regions of G_2. Since the probability of accidentally forming a partition of size $|\Lambda_i|$ in the initial labelling is equal to $V_2^{-|\Lambda_i|}$ we gauge the overall consistency of each partition by the number of nodes contained within it. In practical matching applications the regions of spurious consistency are small, typically much smaller than correctly labelled regions, and in this case it is sufficient to reject those regions whose size falls below some critical size. In this way we can effectively filter-out relational units which are not consistently matched. This constraint filtering process can be regarded as a rudimentary form of graph editing that is applied to locate consistent subgraphs once the optimisation of consistent matches has converged. Good results can therefore be anticipated if the matching process is concerned with locating fragmented subgraphs. If, on the other hand, the structural errors uniformly permeate the graphs then a finer editing strategy is required. In the next section we describe an active graph matching technique which aims to couple the iterative optimisation of consistency to the correction of structural errors using incremental edit operations.

4.3 Active matching with graph-edits

Under conditions of extreme clutter relational fragmentation will severely limit the matching methods described in Sections 4.1 and 4.2. The basic problem originates from structural disruption of the consistent relations residing in the dictionary by the addition of clutter nodes. For instance, if the relational structure being matched is a Delaunay graph representing a Voronoi tessellation of the image plane, then a single additional clutter node may create many additional graph edges [11]. One way of overcoming this problem is to edit-out clutter nodes. In this way the relational structure of the data graph is reconstructed in an active matching process. This process is illustrated in Figure 2.

We have recently shown how this process of active graph reconfiguration can be realised as a MAP estimation process [11]. The basic idea is to gauge the net effect of deleting a node by examining those contributions to the consistency measure that arise from modification of the super-cliques containing the node in question. This set is constructed by identifying those nodes that form a super-clique with node u in graph G_1, i.e. $C_u - \{u\}$, and determining the new super-clique set for these nodes in the reconfigured graph G_1'. We let χ_u^+ denote the

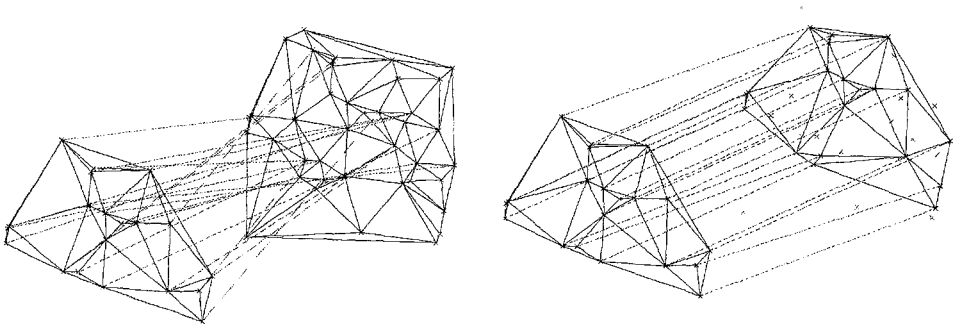


Fig. 2. Original graph (left) and edited graph (right) with their associated matches.

super-clique set of object u in graph G_1 and χ_u^- denote the corresponding super-clique set in the reconfigured graph G_1'. With this notation the change in the MAP criterion caused by the deletion of the node u is proportional to

$$\Delta_u^- = P_\phi \sum_{j \in \chi_u^-} \frac{K_{C_j}}{|\Theta_j|} \sum_{S_i \in \Theta_j} \exp[-k_e H(\Gamma_j, S_i)] \tag{7}$$

By contrast, when considering the change in the MAP criterion caused by re-insertion of the node u it is the super-clique set χ_u^+ to which we turn our attention. The corresponding change to the MAP criterion is proportional to

$$\Delta_u^+ = \frac{P(u, f(u)|\underline{x}_u^1, \underline{x}_{f(u)}^2)}{P(u, f(u))} \sum_{j \in \chi_u^+} \frac{K_{C_j}}{|\Theta_j|} \sum_{S_i \in \Theta_j} \exp[-k_e H(\Gamma_j, S_i)] \tag{8}$$

The decision criteria for node deletion or re-insertion are as follows. We delete node u provided $\Delta_u^+ < \Delta_u^-$ and reinstate it provided $\Delta_u^+ > \Delta_u^-$. Once the graph has been reconfigured or edited in this way, consistency must be restored to those portions of the graphs to have undergone structural modification. Adhering to our MAP philosophy, consistency is maintained provided that the mapping function f is updated according to equation (6). The graph reconfiguration procedure implied by equation (8) and the node relabelling operation of equation (6) effectively perform a function analogous to Sanfeliu and Fu's graph edit operations [6]. Rather than requiring the specification of a series of heuristic cost terms, the node deletions and re-insertions are regulated by the null match probability P_ϕ and the error probability P_e. Since the parameter P_e fulfils the role of a control variable which imposes incremental hardening on the constraints residing in the dictionary, our structural error model has only one free parameter, P_ϕ.

5 Sensitivity Analysis

In order to provide some objective performance measures, we have conducted the bulk of our experiments using synthetic relational graphs. In order to embark on this study, we have seeded Voronoi tessellations from random dot patterns and computed the associated Delaunay graph. The nodes of the graph are therefore the random dots, while the edges indicate that the associated Voronoi regions are adjacent to one-another. In order to simulate the effects of clutter and segmental dropout, we have both added dots at random locations and deleted random dots from the patterns used to seed the Voronoi tessellation. This has the effect of corrupting the topology of the associated Delaunay graph. In fact the addition of random clutter amplifies the number of structural errors. For instance when the fraction of clutter nodes is 10%, approximately 20% of the nodes in any relation are corrupt. When a quarter of the nodes are clutter, then about one-half the edges in each relational unit are spurious. In addition to the structural errors introduced by random dot insertions and deletions, we have also added random initialisation errors. For the majority of the experiments described here the level of initialisation error is approximately 50%.

Figure 3a shows the fraction of the original graph correctly matched as a function of the fraction of added clutter. The solid line represents the result of active matching with graph-edit operations (see Section 4.3). The dashed line is the result of applying constraint filtering to the association graph of the optimal matches (see Section 4.2). Finally, the dotted line is the result of null-labelling (see Section 4.1). It is clear that the active matching method consistently outperforms both null-labelling and constraint filtering. Errors do not appear until the fractional corruption exceeds 20%. Even when the clutter fraction is as high as 50%, then 80% matching accuracy is achievable. By contrast, the two remaining methods are 10%-20% more susceptible to error. However, it is interesting to note that the constraint filtering technique has a tangible performance edge over null-labelling. This is largely attributable to the difficulties encountered in controlling the parameter of the null match. For comparison, Figure 3b shows the fraction of residual noise after matching; this represents the fraction of added noise which is not assigned to the null category. Again, the best performance is obtained using graph-edit operations, with null-labelling delivering the poorest performance.

6 Conclusions

We have compared three different strategies for rectifying structural errors in relational matching. The framework for this study has been furnished by a Bayesian model of relational inexactness. The basic ingredient is a relational consistency criterion which draws on Hamming distance as its basic measurement. This criterion has been exploited to control structural errors in a null-tagging process, a constraint filtering algorithm and a graph-edit technique. Of the three alternatives the graph-edit technique has proved to be most robust to structural errors.

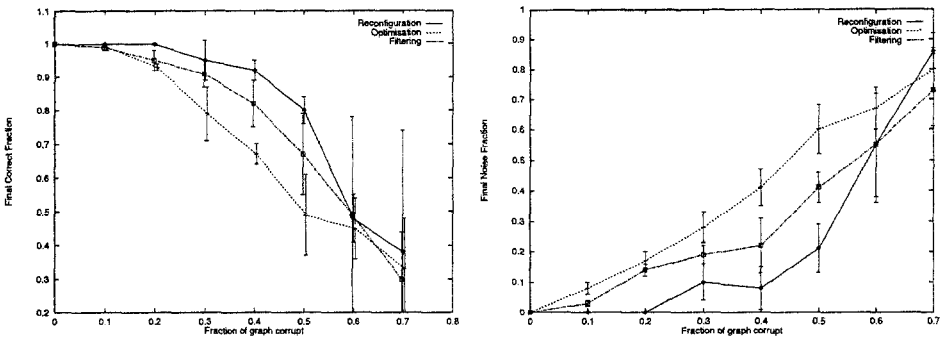


Fig. 3. a) Correctly matched fraction (left) and b) residual noise fraction (right).

References

1. H.G. Barrow and R.J. Popplestone, "Relational Descriptions in Picture Processing", *Machine Intelligence*, **VI**, pp 377–396, 1971.
2. H.G. Barrow and R.M Burstall, "Subgraph Isomorphism, Matching Relational Structures and Maximal Cliques", *Information Processing Letters*, **4**, pp.83–84, 1976.
3. K. Boyer and A. Kak, "Structural Stereopsis for 3D Vision", *IEEE PAMI*, 10, pp 144-166, 1988.
4. D. Geman and S. Geman, "Stochastic Relaxation, Gibbs Distributions and Bayesian Restoration of Images", *IEEE PAMI*, **6**, pp 721-741, 1984.
5. J. Kittler, W.J. Christmas and M.Petrou, "Probabilistic Relaxation for Matching Problems in Machine Vision", *Proceedings of the Fourth International Conference on Computer Vision*, pp. 666-674, 1993.
6. A. Sanfeliu and K.S. Fu, "A Distance Measure between Attributed Relational Graph", *IEEE SMC*, **13**, pp 353–362, 1983.
7. L.G. Shapiro and R.M.Haralick, "Structural Description and Inexact Matching", *IEEE PAMI*, **3**, pp 504–519, 1981.
8. L.G. Shapiro and R.M.Haralick, "A Metric for Comparing Relational Descriptions", *IEEE PAMI*, **7**, pp 90-94, 1985.
9. R.C. Wilson and E.R Hancock, "Graph Matching by Discrete Relaxation", *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, North Holland, pp. 165–177, 1994.
10. R.C. Wilson and E.R Hancock, "Relational Matching by Discrete Relaxation", *Image and Vision Computing*, **13**, pp. 411–422, 1994.
11. R.C. Wilson and E.R. Hancock, "Relational Matching with Dynamic Graph Structures" *Proceedings of the Fifth International Conference on Computer Vision*, pp. 450–456, 1995.

Parallel Distributed Relaxation Labeling

S. Z. Li

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
szli@szli.eee.ntu.ac.sg

Abstract. A parallel distributed relaxation labeling (RL) method, called the Lagrange-Hopfield (LH) method, is presented. RL is treated as a constrained optimization problem. The LH method solves the problem using the augmented Lagrangian multiplier technique and the graded Hopfield network. The LH method effectively overcomes instabilities that are inherent in the penalty method (*e.g.* Hopfield network) or the Lagrange multiplier method in constrained optimization. Due to the use of Lagrangian multipliers, the normalization operation in traditional RL methods is dispensed with. This makes the LH algorithm fully parallel and distributed and is suitable for analog implementation. Experiments also show that the method is able to produce good solutions in terms of the optimized objective values.

1 Introduction

Relaxation labeling (RL) [17] is a class of parallel iterative numerical procedures which use contextual constraints to reduce ambiguities in image analysis. It is widely used to solve image processing and vision problems where contextual constraints play an important role. The applications include image smoothing, enhancement, remote sensing, edge and boundary segmentation, motion analysis, perceptual organization and object matching and recognition.

Mathematically, RL can be posed as a problem of constrained optimization, as pointed out by Ullman [18]. Faugeras and Berthod [4] define a class of global criteria in terms of transition probabilities. Haralick [7] illustrates RL as minimizing the expected loss from a Bayes viewpoint. Hummel and Zucker [10] formulate the probabilistic relaxation approach as a global optimization problem in which a so called average local consistency measure is maximized.

There are various algorithms which can be used for computing RL solutions. Two basic types of RL algorithms are proposed in [17] for discrete and continuous updating of labeling assignment. A method known as gradient projection is used for the constrained optimization in [4, 10], which are shown in a comparison work [16] to yield good quality solutions. Techniques from mean field theory may be applied to obtain approximate solutions [13]. The above are deterministic RL algorithms which are local optimizers. A stochastic RL algorithm, simulated annealing (SA) [11, 5], is proven to converge a global optimum when the annealing

* This work is supported by NTU project ARC-1/94.

schedule is slow enough. Disadvantages of SA are the high computational cost and difficulties in tuning the annealing schedule [1].

Although the continuous RL algorithms such as [17, 4, 10, 13] are parallelizable, they are not distributive. This is because of the requirement that the labeling assignment for each object to be labeled be normalized, *i.e.* summed into 1. The gradient projection in [4, 10] is actually a global operation.

This paper proposes a novel algorithm, called the Lagrange-Hopfield (LH) method, for continuous RL. It is based on the augmented Lagrange multiplier method for constrained optimization [15, 8] and the graded Hopfield method [9] for bridging the gap between discrete and continuous optimization. As an optimization network, the LH method effectively overcomes instabilities that are inherent in the penalty method (*e.g.* Hopfield network) or the Lagrange multiplier method. Owing to the use of Lagrangian multipliers, the normalization operation in traditional RL methods whereby the labeling assignment for each object is summed into one is dispensed with. The resulting algorithm solves a system of unconstrained differential equations. This makes the LH algorithm fully parallel and distributed and is suitable for analog implementation.

Experiments are done to compare the LH algorithm with the deterministic RL algorithms of [17, 10, 13] and the SA algorithm of [11]. The results show that the LH algorithm yields good quality solutions comparable to produced by simulated annealing.

The rest of the paper is organized as follows: Section 2 introduces the notations for RL. Section 3 presents the LH method. Section 4 describes the experiments. Section 5 gives conclusions.

2 Continuous Relaxation Labeling

Let $\mathcal{S} = \{1, \ldots, m\}$ be a set of m sites (termed objects in the traditional RL literature) to be labeled and $\mathcal{L} = \{0, 1, \ldots, M\}$ be a set of $M + 1$ labels to be assigned to the sites. In continuous RL, the labeling state for each site $i \in \mathcal{S}$ is represented by an $M + 1$ position vector

$$p_i = [p_i(I) \mid I \in \mathcal{L}] \tag{1}$$

subject to the following *feasibility* constraints

$$\begin{cases} \sum_{I \in \mathcal{L}} p_i(I) = 1 \ \forall i \\ p_i(I) \geq 0 \qquad \forall i, I \end{cases} \tag{2}$$

The real value $p_i(I) \in [0, 1]$ reflects the strength with which i is assigned label I. Each p_i lies in a hyperplane in the non-negative quadrant portion of the multi-dimensional real space $\mathbb{R}^{\mathcal{L}}$. The set $p = \{p_i \mid i \in \mathcal{S}\}$ is called a *labeling assignment*. The feasible space for labeling assignments is

$$\mathbb{P} = \{p \mid p_i(I) > 0, \sum_{I \in \mathcal{L}} p_i(I) = 1, i \in \mathcal{S}, I \in \mathcal{L}\} \tag{3}$$

However, the final solution p^* must be *unambiguous*, that is,

$$p_i^*(I) = \{0, 1\} \qquad \forall i, I \tag{4}$$

with $p_i^*(I) = 1$ meaning that i is unambiguously labeled I. Therefore, the space for the final labeling assignments is

$$\mathbb{P}^* = \{p^* \mid p_i(I) \in \{0,1\}, \sum_{I \in \mathcal{L}} p_i(I) = 1, i \in \mathcal{S}, I \in \mathcal{L}\} \tag{5}$$

$\mathbb{P}^*$ is the set of "corners" of $\mathbb{P}$. Non-corner points in the continuous space $\mathbb{P}$ provide routes to the corners, *i.e.* the points in $\mathbb{P}^*$.

The continuous RL can be regarded as an iterative process for minimizing the following energy

$$\begin{aligned} E(p) = & \sum_{i \in \mathcal{S}} \sum_{I \in \mathcal{L}} r_i(I)\, p_i(I) + \\ & \sum_{i \in \mathcal{S}} \sum_{I \in \mathcal{L}} \sum_{i' \in \mathcal{S}, i' \neq i} \sum_{I' \in \mathcal{L}} r_{i,i'}(I, I')\, p_i(I)\, p_{i'}(I') \end{aligned} \tag{6}$$

where $r_i(I)$ and $r_{i,i'}(I, I')$ are unary and binary compatibility functions. The minimization is over $\mathbb{P}$ and finally over $\mathbb{P}^*$ and hence is constrained. The constrained minimization is done based on the gradient $q = \nabla E(p) = \{q_i(I) \mid \forall (i, I)\}$ where the gradient components are given by

$$q_i(I) = \frac{\partial E}{\partial p_i(I)} = r_i(I) + 2 \sum_{i' \in \mathcal{S}, i' \neq i} \sum_{I' \in \mathcal{L}} r_{i,i'}(I, I')\, p_{i'}(I') \tag{7}$$

for symmetric $r_{i,i'}(I, I')$. The initial assignment is set to $p^{(0)} \in \mathbb{P}$. The current $p^{(t)}$ is updated into a new state $p^{(t+1)}$ according to

$$p^{(t+1)} \leftarrow \Phi(p^{(t)}, q^{(t)}) \tag{8}$$

where Φ is an updating operator. RL algorithms differ from one another in the choice of Φ.

3 Lagrange-Hopfield Method

Let us state the RL problem more formally as the following constrained optimization:

$$\min_{f} \quad E(p) \tag{9}$$

$$\text{subject to} \quad C_i(p) = 0 \quad \forall i \in \mathcal{S} \tag{10}$$

$$p_i(I) \geq 0 \quad \forall i \in \mathcal{S}, I \in \mathcal{L} \tag{11}$$

where

$$C_i(p) = \sum_{I \in \mathcal{L}} p_i(I) - 1 \tag{12}$$

Equation (10) and (11) defines the feasible space $\mathbb{P}$. The final solution must be unambiguous:

$$p_i^*(I) \in \{0, 1\} \tag{13}$$

The LH method for solving the above problem is described in the following.

First, we introduce internal variables $u_i(I) \in (-\infty, +\infty)$ $(\forall i, I)$ and relate them to $p_i(I)$ via

$$p_i(I) = \psi_T(u_i(I)) \tag{14}$$

where $\psi_T(x)$ is a sigmoid function

$$\psi_T(x) = 1/[1 + e^{-x/T}] \tag{15}$$

controlled by a temperature parameter $T > 0$. With the introduction of the u variables, the energy function can be considered as $E(u) = E(p(u))$. This treatment confines $p_i(I)$ to the range $(0, 1)$ and imposes the constraints of (11). In very high gain when $T \to 0^+$, $p_i(I)$ is forced to be 0 or 1 depending on whether $u_i(I)$ is positive or negative, thus imposing the constraints of (13).

Next, we use the Lagrange multiplier method to impose the constraints of (10). Define the following Lagrange function

$$L(p, \gamma) = E(p) + \sum_{i \in \mathcal{S}} \gamma_i C_i(p) \tag{16}$$

where γ_i are called Lagrange multipliers. It is a function of the $M \times m$ variables of p and m variables of γ. Penalty terms $[C_i(p)]^2$ can be added to give an augmented Lagrange function [15, 8]

$$L(p, \gamma) = E(p) + \sum_i \gamma_i C_i(p) + \beta \sum_i [C_i(p)]^2 \tag{17}$$

where $\beta > 0$ is a finite weight. Our experiments show that the penalty terms are generally necessary for damping oscillations and ensuring the convergence for the algorithm described below.

For p^* to be a local minimum subject to the constraints, it is necessary that (p^*, γ^*) be a stationary point of the Lagrange function:

$$\begin{aligned} \nabla_p L(p^*, \gamma^*) &= 0 \\ \nabla_\gamma L(p^*, \gamma^*) &= 0 \end{aligned} \tag{18}$$

If (p^*, γ^*) is a saddle point for which

$$L(p^*, \gamma) \leq L(p^*, \gamma^*) \leq L(p, \gamma^*) \tag{19}$$

then p^* is a local minimum of $E(p)$ satisfying $C_k(p^*) = 0$ [6].

The following dynamics for the p variables can be used to find such a saddle point

$$\begin{aligned}\frac{\mathrm{d}p_i(I)}{\mathrm{d}t} &= -\frac{\partial L(p,\gamma)}{\partial p_i(I)} \\ &= -\frac{\partial E(p)}{\partial p_i(I)} - \gamma_i \frac{\partial C_i(p)}{\partial p_i(I)} - \beta C_i(p)\frac{\partial C_i(p)}{\partial p_i(I)}\end{aligned} \tag{20}$$

$$\frac{\mathrm{d}\gamma_i}{\mathrm{d}t} = +\frac{\partial L(p,\gamma)}{\partial \gamma_i} = +C_i(p) \tag{21}$$

where $\frac{\partial E(p)}{\partial p_i(I)} = q_i(I)$ is defined in (7) and $\frac{\partial C_i(p)}{\partial p_i(I)} = 1$ for $C_i(p) = \sum_{I\in\mathcal{L}} p_i(I) - 1$. It performs energy descent on p but ascent on γ. The convergence of this system has been obtained by [2]. Its application in relaxation labeling is suggested by [18] without using the u variables. It has also been used for solving the traveling salesman problem [14, 19].

In the LH method, the updating of the labeling variables is performed on u rather than on p. Equ.(20) is replaced by

$$\begin{aligned}\frac{\mathrm{d}u_i(I)}{\mathrm{d}t} &= -\frac{\partial L(p(u),\gamma)}{\partial p_i(I)}\frac{\partial p_i(I)}{\partial u_i(I)} \\ &= -\left[q_i(I)/m + \gamma_i + \beta C_i(p)\right]\frac{\partial p_i(I)}{\partial u_i(I)}\end{aligned} \tag{22}$$

where a $q_i(I)$ is weighted by a factor of $1/m$ to achieve some invariance to m. Note that $\frac{\partial p_i(I)}{\partial u_i(I)} = \frac{\mathrm{e}^{-u_i(I)/T}}{T(1+\mathrm{e}^{-u_i(I)/T})^2}$ is always positive.

A damping term $\zeta u_i(I)$, where $\zeta > 0$, may be added to (22) as in the graded Hopfield neural networks [9]. This is equivalent to adding an energy term

$$H(p) = \sum_i \sum_I \int_0^{p_i(I)} \psi_T^{-1}(p)\mathrm{d}p \tag{23}$$

to $L(p(u),\gamma)$, where the inverse of the sigmoid function ψ_T is

$$\psi_T^{-1}(p_i(I)) = u_i(I) = T\ln\frac{p_i(I)}{1-p_i(I)} \tag{24}$$

The term $H(p)$ reaches the minimum of zero only when all $p_i(I)$'s are either 0 or 1. This means minimizing this term with $T \to 0^+$ in effect leads to the satisfaction of (13). However, this term is not necessary in our model because the satisfaction of (10) and (11) under a small T for $p_i(I) = 1/[1+e^{-u_i(I)/T}]$ is enough for the convergence to (13). The competitive mechanism in (11) will make the winner take all.

In summary, the LH algorithm implements the three equations, (21), (14) and (22). The updating consists of the three steps:

$$\begin{aligned}&u_i(I) \leftarrow u_i(I) - \\ &\mu\tfrac{\partial p_i(I)}{\partial u_i(I)}\left\{q_i(I)/m + \gamma_i + \beta\left[\textstyle\sum_I p_i(I) - 1\right]\right\}\end{aligned} \tag{25}$$

where μ is a step size factor

$$p_i(I) \leftarrow \psi_T(u_i(I)) \tag{26}$$

and

$$\gamma_i \leftarrow \gamma_i + \lambda \left[\sum_I p_i(I) - 1 \right] \tag{27}$$

where T is decreasing and λ increasing. The above is formulated for minimizing the energy function $E(p)$. To maximize a function $G(f)$ with the constraints, we can simply let $E(p) = -G(p)$ and then apply the same updating equations.

The above LH algorithm represents an unconstrained system and it is in contrast to the existing RL algorithms. RL needs a normalization process or alike to meet the feasibility condition. In [17], for example, Rosenfeld *et al.* use the following fixed point iteration

$$p_i^{(t+1)}(I) \leftarrow \frac{p_i^{(t)}(I)(1 + q_i^{(t)}(I))}{\sum_I p_i^{(t)}(I)(1 + q_i^{(t)}(I))} \tag{28}$$

for maximizing E(p). Other improved algorithms involve more sophisticated operations such as gradient projection [4, 10] and mean field annealing [13]. These are dispensed with in LH due to the use of the Lagrange multipliers. The way that LH works make it suitable for distributive implementation.

4 Experiments

In the following experiments, we compare several RL algorithms in terms of the solution quality measured by the maximized gain $G(p^*)$, the computational cost measured by the number of required iterations, and the need for heuristics for tuning the algorithm. The compared algorithms are

1. ICM [3]
2. RL of [10]
3. RL of [17]
4. MFA of [13]
5. SA of [5]
6. The LH method

The test-bed for the comparison is relational matching. The MAP-MRF formulation described in [12] is used to define the compatibility matrix $[r_{i,i'}(I, I')]$. The parameters in Equations (25) – (27) are set as follows: $\beta = 100, T$ decreasing from 100000 to not smaller than 10000 according to $T \leftarrow 0.999T$, λ increasing from 1 to 100 according to $\lambda \leftarrow 1.01\lambda$, and $\mu \frac{\partial p_i(I)}{\partial u_i(I)} = 1000$ is constant for the sequential implementation. The variables are initialized as $u_i(I) = 0$ and $\gamma_i = 0$. Figures 1 and 2 demonstrate the solution quality in terms of the maximized gain and the cost in terms the number of iterations, respectively. Every result is due to the average of 200 runs.

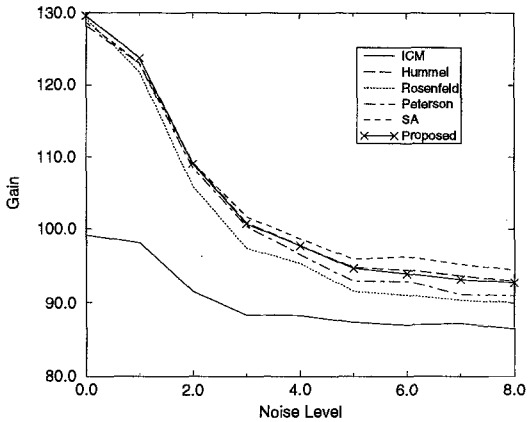


Fig. 1. The maximized gain $G(p^*)$ (after divided by a factor of m^2). The higher, the better.

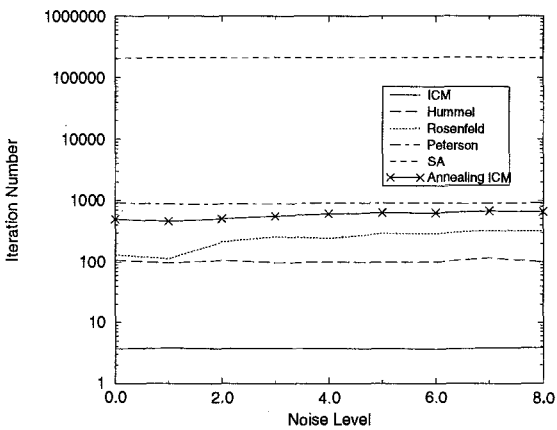


Fig. 2. The number of iterations. The lower, the better.

In terms of the solution quality, the proposed RL algorithm performs almost as well as the benchmarking SA algorithm and the best of the existing RL algorithms. In terms of the iteration number, it is several order lower that for SA. Further, it is much simpler than SA in terms of the ease in the tuning of parameters and schedules.

5 Conclusion

A novel relaxation labeling (RL) method, called the Lagrange-Hopfield (LH) method, has been proposed. The stabilities of the LH are due to the use of the augmented Lagrange technique. Basically, one is converting a combinatorial optimization to a constrained real optimization and in turn to an unconstrained

real optimization. In the last form, the energy for the unconstrained problem is a weighted sum of the original cost function and other terms for the constraints. To produce valid solutions, the relative weight for the original cost function must be kept small to verify the constraints; therefore the solutions are significantly biased from the true minimum of the original cost function, owing to large influence from the extra terms. With the use of Lagrange multipliers as in the augmented Lagrangian, the weighting values for the penalty terms can be much smaller than those required by the penalty method; thus the relative weight for the original objective is increased, which helps yielding a lower objective value. At the same time, the augmented Lagrange method also overcomes the zigzagging problem with the standard Lagrange method and improve the convergence. These advantages of the augmented Lagrangian plus the use of Hopfield method for imposing the inequality and unambiguity constraints have made the success of the LH method.

The LH is comparable to the best of the existing RL algorithms in terms of solution quality and convergence rate. Its advantage over the existing RL algorithms is that it avoids the needs for the normalization and/or gradient projection operations. The dynamic system for the computation of LH solutions is suitable for parallel distributed implementation using analog circuits. Such hardware is useful not only for problems which can be formulated as relaxation labeling such as edge labeling and object matching; but also for general combinatorial problems such as traveling sales-man problem, pattern clustering and perceptual grouping.

References

1. E. H. L. Aarts. *Simulated annealing and Boltzmann machines : a stochastic approach to combinatorial optimization and neural computing.* Wiley, 1989.
2. K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Nonlinear Programming.* Stanford University Press, 1958.
3. J. Besag. "On the statistical analysis of dirty pictures" (with discussions). *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.
4. O. D. Faugeras and M. Berthod. "Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:412–423, April 1981.
5. S. Geman and D. Geman. "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.
6. B. S. Gottfried. *Introduction to Optimization Theory.* Prentice-Hall, 1973.
7. R. M. Haralick. "Decision making in context". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):417–428, July 1983.
8. M. R. Hestenes. "Multipler and gradient methods". *Journal of Optimization Theory and Applications*, 4:303–320, 1969.
9. J. J. Hopfield. "Neurons with graded response have collective computational properties like those of two state neurons". *Proceedings of National Academic Science, USA*, 81:3088–3092, 1984.

10. R. A. Hummel and S. W. Zucker. "On the foundations of relaxation labeling process". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–286, May 1983.
11. S. Kirkpatrick, C. D. Gellatt, and M. P. Vecchi. "Optimization by simulated annealing". *Science*, 220:671–680, 1983.
12. S. Z. Li. "A Markov random field model for object matching under contextual constraints". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 866–869, Seattle, Washington, June 1994.
13. C. Peterson and B. Soderberg. "A new method for mapping optimization problems onto neural networks". *International Journal of Neural Systems*, 1(1):3–22, 1989.
14. J. C. Platt and A. H. Barr. "Constrained differential optimization". In *Proceedings of the IEEE 1987 NIPS conference*, 1988.
15. M. J. D. Powell. "A method of nonlinear constraints in minimization problems". In R. Fletcher, editor, *Optimization*, London, 1969. Academic Press.
16. K. E. Price. "Relaxation matching techniques – A comparison". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5):617–623, September 1985.
17. A. Rosenfeld, R. Hummel, and S. Zucker. "Scene labeling by relaxation operations". *IEEE Transactions on Systems, Man and Cybernetics*, 6:420–433, June 1976.
18. S. Ullman. "Relaxation and constraint optimization by local process". *Computer Graphics and Image Processing*, 10:115–195, 1979.
19. E. Wacholder, J. Han, and R. C. Mann. "A neural network algorithm for the multiple traveling salesman problem". *Biological Cybernetics*, 61:11–19, 1989.

Subgraph Isomorphism Detection in Polynomial Time on Preprocessed Model Graphs

B.T. Messmer and H. Bunke

Institut für Informatik und angewandte Mathematik, University of Bern,
Neubrückstr. 10, CH-3012 Bern, Switzerland,
messmer@iam.unibe.ch, bunke@iam.unibe.ch

Abstract. In this paper, a new method for graph and subgraph isomorphism detection based on a decision tree representation is proposed. The decision tree is generated off-line from a priori known model graphs. At run time the decision tree is used to detect all graph and subgraph isomorphisms from an input graph to any of the model graphs in time that is only polynomial in the size of the graphs and independent of the number of model graphs. However, the decision tree is of exponential size. In order to reduce the size of the decision tree, we propose two pruning techniques. Experimental results confirming the efficiency of the method will be given.

1 Introduction

Graph and subgraph isomorphism are concepts that have been intensively used in various applications. Due to their representational power, graphs are especially useful for applications in the domain of computer vision and pattern recognition. Subgraph isomorphism detection is used to recognize and locate 2D or 3D objects (see [1] for an overview). Different techniques for the computation of graph and subgraph isomorphism have been proposed in the past, such as backtracking and discrete relaxation[2, 3], or maximal clique detection in an association graph[4]. These methods are guarantueed to find all subgraph isomorphisms between two graphs. However, due to the exponential nature of the problem, they may require exponential time in the worst case. For the special case of graph isomorphism detection, several methods of polynomial complexity have been proposed [5, 6]. However, they generally impose strong restrictions on the underlying graphs.

An additional problem in most practical applications is that normally there is a set of several model graphs that must be matched via graph or subgraph isomorphism with an unknown input graph. In a straightforward implementation, each of these model graphs is matched individually with the input graph by applying one of the above mentioned methods. However, if the number of a priori known model graphs is large then the sequential testing of each model graph becomes computationally very costly.

In this paper, we propose a method which is capable of finding all subgraph isomorphisms between an input graph and a set of model graphs in time that is only quadratic in the number of vertices of the input graph. In particular, the

time complexity of the new method is completely independent of the number of model graphs in the database. The new algorithm mainly relies on an intensive preprocessing step. In this preprocessing step, the adjacency matrices of the model graphs in the database are compiled into a decision tree. At run time, the decision tree is used to classify the adjacency matrix of an unknown input graph. The main disadvantage of the new method is the size of the decision tree. In the worst case, the number of nodes in the decision tree grows exponentially with the number of vertices of the model graphs. In order to make the new approach useful for practical applications, we developed a number of pruning techniques. The new method has been described in detail in [7].

2 Preliminaries

A graph G is a 3-tuple $G = (M, L_v, L_e)$ where M is an adjacency matrix with $M = (m_{ij}); i, j = 1, \ldots, n$; $m_{ii} \in L_v$, $m_{ij} \in L_e$ for $i \neq j$; L_v and L_e are sets of labels. The diagonal of the matrix M represents the labels of the vertices of G, while the other entries in M represent edge labels. Clearly, the matrix M is not unique for a graph G. If M represents G, then any permutation of M is also a valid representation of G. An $n \times n$-matrix $P = (p_{ij})$ is called a *permutation matrix* if $p_{ij} \in \{0, 1\}$, $\sum_{i=1}^{n} p_{ij} = 1$ and $\sum_{j=1}^{n} p_{ij} = 1$ for $i, j = 1, \ldots, n$. If a graph G is represented by an $n \times n$ adjacency matrix M and P is an $n \times n$ permutation matrix, then the $n \times n$ matrix $M' = PMP^T$ where P^T denotes the transpose of P, is also an adjacency matrix of G. With this notation, it is possible to define the concept of graph isomorphism. Two graphs $G_1 = (M_1, L_v, L_e)$ and $G_2 = (M_2, L_v, L_e)$ are *isomorphic* if there exists a permutation matrix P such that

$$M_2 = PM_1P^T \tag{1}$$

Given an $n \times n$ matrix $M = (m_{ij})$, let $S_{k,m}(M)$ denote the $k \times m$ matrix that is obtained from M by deleting rows $k+1, \ldots, n$ and columns $m+1, \ldots, n$, where $k, m \leq n$. A *subgraph* S of a graph G, $S \subset G$, is a graph $S = (M', L_v, L_e)$ where $M' = S_{m,m}(PMP^T)$ is an $m \times m$ adjacency matrix for some permutation matrix P. The concept of subgraph isomorphism can now be described as follows. Let G_1 and G_2 be graphs with adjacency matrices M_1 and M_2 of dimensions $m \times m$ and $n \times n$ respectively, where $m \leq n$. There is a *subgraph isomorphism* from G_1 to G_2 if there exists an $n \times n$ permutation matrix P such that

$$M_1 = S_{m,m}(PM_2P^T) \tag{2}$$

Thus, the problem of finding a subgraph isomorphism from G_1 to G_2 is equivalent to finding a permutation matrix P for which (2) holds.

3 Subgraph Isomorphism by Means of Decision Trees

We assume that there is a set of model graphs that are known a priori, while the input graph becomes accessible at run time only. We intend to find all graph and

subgraph isomorphisms from the input graph to the model graphs. A conventional algorithm such as Ullman's algorithm will try to compute all permutation matrices that satisfy Eq. 2 at run time. Because there may be an exponential number of such permutation matrices, the conventional scheme will take exponential time in the worst case. We now propose an entirely different approach.

For each model graph we compute off-line all possible permutations of its adjacency matrix and transform these adjacency matrices into a decision tree. At run time, the matrix of the input graph is then used to find those adjacency matrices in the decision tree that are identical to it. The permutation matrices that correspond to these adjacency matrices represent the graph or subgraph isomorphisms that we are looking for. Let $G = (M, L_v, L_e)$ be a model graph and $A(G)$ denote the set of all permuted adjacency matrices of G, $A(G) = \{M_P | M_P = PMP^T$ where P is a $n \times n$ permutation matrix$\}$. We are now ready to restate the subgraph isomorphism problem in terms of the set introduced above. For a model graph G with corresponding $n \times n$-adjacency matrix M and an input graph G_I with an $m \times m$-adjacency matrix M_I and $m \leq n$, we have to determine whether there exists a matrix $M_P \in A(G)$ such that $M_I = S_{m,m}(M_P)$. If such a matrix M_P exists, the permutation matrix P corresponding to M_P describes a subgraph isomorphism from G_I to G, i.e. $M_I = S_{m,m}(M_P) = S_{m,m}(PMP^T)$ (see Eq. 2). If G and G_I are of equal size, the permutation matrix P represents a graph isomorphism between G_I and G, i.e. $M_I = PMP^T$.

We propose to organize the set $A(G)$ in a decision tree such that each matrix in $A(G)$ is classified by the tree. The features that will be used for the classification process are the individual elements in the adjacency matrices. In order to classify matrices of different size, we group the individual elements of a matrix into size independent features. We say that the matrix consists of an array of so-called *row-column* elements a_i, where each a_i is a vector of the form $a_i = (m_{1i}, m_{2i}, \ldots, m_{ii}, m_{i(i-1)}, \ldots, m_{i1})$. The matrix can then be written as $M = (a_1, a_2, \ldots, a_n)$; $i = 1, \ldots, n$. The decision tree is now built according to the row-column elements of each adjacency matrix $M_P \in A(G)$. At the top of the decision tree there is a single root node. The direct successor nodes of the root node constitute the first level of the decision tree. On the first level, the classification of the matrices in $A(G)$ is done according to the first row-column element a_1 of each matrix $M_P \in A(G)$. The element $a_1 = (m_{11})$ represents the label of the first vertex in each matrix in $A(G)$, with $m_{11} \in L_v$ (see Section 2). Consequently, each matrix in $A(G)$ is classified according to its first vertex label. Each branch that leads to a direct successor node of the root node is associated with a specific value for the row-column element a_1. Next, on the second level of the decision tree, the second row-column element a_2 of each matrix is used for the classification, and so on. In general, the matrices that are represented by some node on the level k are divided into classes according to the element a_k.

With each matrix M_P that is represented by some node N on the level k, the corresponding permutation matrix P is also given. As M_P has been classified up to the k-th vertex, P describes a subgraph isomorphism for the subgraph $S_{k,k}(M_P)$ to G. Finally, at the bottom of the decision tree, there are the leaf

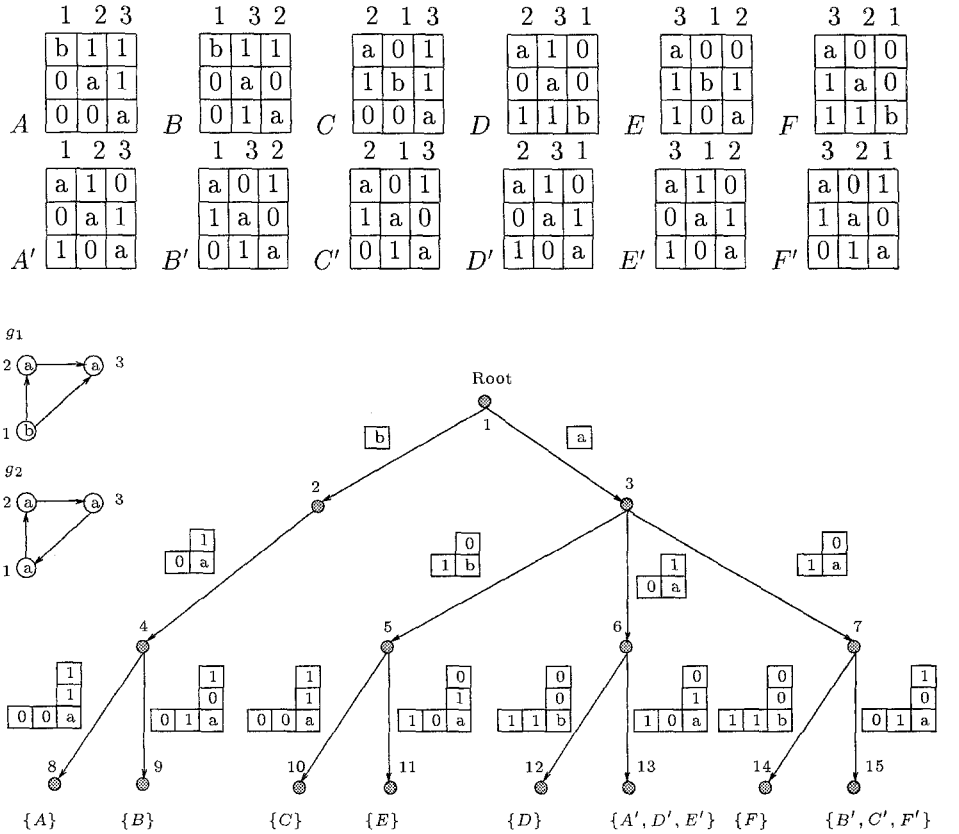


Fig. 1. Decision tree for the classification of the adjacency matrices $A \ldots F$ and $A', \ldots, F'$ of the graphs g_1 and g_2, respectively.

nodes. Each leaf node N represents a class of identical matrices $M_P \in A(G)$.

So far the construction of a decision tree for a single model graph has been explained. However, it is easy to see that different model graphs can be represented by the same decision tree. The classification of an adjacency matrix by the decision tree is independent of the model graph to which the adjacency matrix corresponds. Consequently, a set of different model graphs $G_1, \ldots, G_L$ and their corresponding sets of adjacency matrices $A(G_1), \ldots, A(G_L)$ can be represented by the same decision tree.

In Fig. 1 two graphs, g_1 and g_2, and the corresponding decision tree is shown. The nodes of the decision tree are represented by shaded circles. Each directed branch from one node to another has associated with it a row-column element. At the top of Fig. 1 the set $A(g_1)$ of permuted adjacency matrices of g_1 is listed. Below, the set $A(g_2)$ of adjacency matrices of g_2 is also given. Each of the matrices in $A(g_1)$ and $A(g_2)$ is classified by the decision tree and represented by some leaf node.

At run time, the decision tree is directly used in order to classify the $m \times m$ adjacency matrix M_I of an unknown input graph G_I. The matrix M_I is classified on the first level according to its row-column element a_{1_I}. If there is some branch i from the root node to a successor node whose associated element a_{1_i} matches a_{1_I}, the algorithm continues with the successor node on the second level and so on. If at some point no classification is possible, then the input graph G_I is not isomorphic to any subgraph of the model graphs or any of the model graphs in the database. If each row-column element of G_I has been used in the classification process and some node N in the decision tree has been reached, then each permutation matrix that is associated with N represents a subgraph isomorphism from the input graph to one of the model graphs. If node N is a leaf node and the input graph and the model graph are of equal size then a graph isomorphism between the input graph and one of the model graphs has been found.[1] For a more detailed description of this algorithm see [7].

The most important aspect of the subgraph isomorphism detection based on a decision tree is that no backtracking is necessary at any point of the decision tree traversal. On each level k, the process of deciding which successor node to follow can be performed in $O(k)$ steps. This is due to the fact that the row-column elements that are associated to the successor nodes of a node N can be stored in a dictionary. This dictionary is organized as a $2k-1$ index structure, in which each element m_{ij} of a row-column element $a_k = (m_{1k}, \ldots, m_{k1})$ is used as an index. Consequently, at run time, the dictionary, which is attached to the node N, is used to look up the existence of a row-column element in $O(k)$ steps. In particular, the computation of the next successor node is independent of the number of successor nodes and therefore independent of the number of model graphs that are represented by the decision tree. For example, consider the successor nodes of the node 4 in Fig. 1. To each branch leading to one of the successor nodes, a row-column element with five entries is associated. There are $2^5 = 32$ different row-column elements possible (although in this example, only two are used). If these elements are organized in a dictionary with five indexes, the comparison of a new row-column element at run time can be performed in exactly five steps. It is easy to imagine more model graphs being added to the decision tree in Fig. 1. However, the number of steps in the node 4 will always be limited to five. We conclude that given a decision tree representing a set of model graphs $G_1, \ldots, G_L$ and an input graph with n vertices, the theoretical run time complexity of the new algorithm is $O(n^2)$ as there are $O(n)$ levels in the decision tree and on each of these levels $O(n)$ steps for choosing the next successor node must be performed. Notice that there is no best or worst case for the new algorithm, i.e., its computation time is always quadratic, while

[1] It is important to note that if the input graph is larger than the model graphs then the decision tree approach cannot not be applied. Consequently, the new method can only be used for the detection of subgraph isomorphisms from the input graph to the model graphs but not for the detection of subgraph isomorphisms from the model graphs to the input graph.

for conventional algorithms such as Ullman's algorithm the computation time becomes exponential in the worst case.

The construction of a decision tree for a given set of model graphs is an off-line process and thus its computational complexity does not affect the run time performance of the subgraph isomorphism algorithm. However, it is easy to see that for a graph with n vertices, there are $n!$ different permuted adjacency matrices and, at worst, $n!$ leaf nodes in the corresponding decision tree. Consequently, the size of the decision tree is bounded by $O(n^n)$ for a single model graph and by $O(Ln^n)$ for L model graphs. As the decision tree must be present in the main memory at run time, its exponential size is prohibitive for most applications. Therefore, in order to make the new method useful for practical applications, the size of the decision tree must be reduced.

4 Reducing the Size of the Decision Tree

In this section, we present two different techniques for the reduction of the number of nodes in a decision tree. The first technique does not change the run time properties of the decision tree in any way. The second technique, however, produces decision trees which can only be used for the detection of graph isomorphisms in polynomial time, but no longer for subgraph isomorphism. Not documented in this paper is a third technique which prunes the depth of the decision trees, thus allowing relatively large graphs to be handled by the new algorithm. For a detailed description of each of the pruning techniques, including the depth-pruning technique, see again [7].

4.1 Redundancy-pruning

In Section 3, the decision tree for a graph G was constructed such that the complete set $A(G)$ of permuted adjacency matrices can be classified and each matrix in $A(G)$ is represented by exactly one leaf node. As a consequence, any subgraph $S_k \subset G$ with k vertices is represented by exactly $k!/\alpha$ nodes, where α denotes the number of automorphisms[2] of S_k. Clearly, the information stored in each of these nodes is redundant. Therefore it is sufficient to choose one node N out of the set of nodes that represent S_k and continue the classification only for the matrices stored in N. All other nodes representing S_k are then redirected to the node N by inserting so-called *redirecting edges*. More formally, given two nodes N and N' representing a subgraph $S_k \subset G$ and the adjacency matrices $M, M' \in A(S_k)$ classified in N and N', there exists a permutation matrix R such that $M = RM'R^T$. Hence, any matrix represented by N' can be transformed into a matrix that is represented by N. The redirecting edge that is inserted between N' and N is associated with the matrix R. At run time, any matrix that is classified by the node N' is transformed by the matrix R and the decision tree traversal is continued via the redirecting edge at node N. Consequently, all

[2] An automorphism is an isomorphism of a graph onto itself.

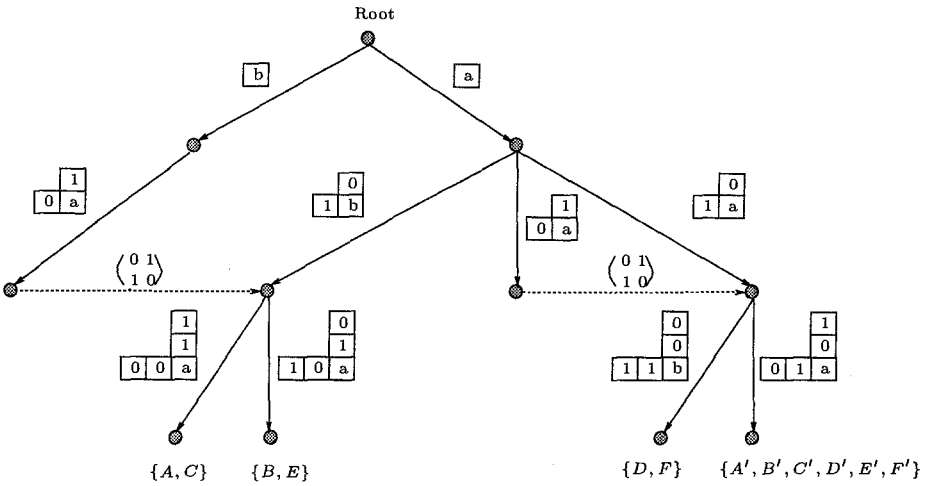


Fig. 2. Decision tree from Fig. 1 after redundancy-pruning was applied.

the successor nodes of N' that are built in the original version of the decision tree can now be discarded.

The number of nodes that can be saved by redundancy-pruning is remarkable. On each level k of the decision tree for a graph with n vertices, the number of nodes is at most $O(\binom{n}{k})$ instead of $O(\binom{n}{k}k!)$ as in the complete decision tree. Thus, the size of the new decision tree is bounded by $O(L3^n)$ instead of $O(Ln^n)$ as for the complete decision tree. For example, in Fig. 2 the decision tree of Fig. 1 is given after redundancy-pruning was applied. Note that four nodes (8,9,12,13) have been pruned by this technique, and redirecting edges are inserted from node 4 to node 5 and from node 6 to node 7. To each of these redirecting edges, a permutation matrix $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is associated which transforms the matrices represented by node 4 and node 6 into the matrices represented by node 5 and node 7, respectively. The redundancy-pruned decision tree can applied for both graph and subgraph isomorphism detection from the input graph to the model graphs. The run time complexity of the algorithm based on a redundancy-pruned decision tree is identical to the complexity for a complete decision tree.

4.2 Breadth-pruning

The second technique for pruning a decision tree is based on the following observation. Let $M_I = (a_1, \ldots, a_m)$ be the adjacency matrix of an input graph G_I. Assume that M_I is to be classified by a decision tree that was built for a model graph G. Furthermore, assume that M_I has been classified up to the level $k+1$ and the node N is the current node in decision tree. In the next step, the algorithm will try to find a successor node of N according to the $k+1$-th element of M_I, a_{k+1}. Note that in the original version of the decision tree there may be $n-k$ possible successor nodes of the current node, namely, for each of the $n-k$

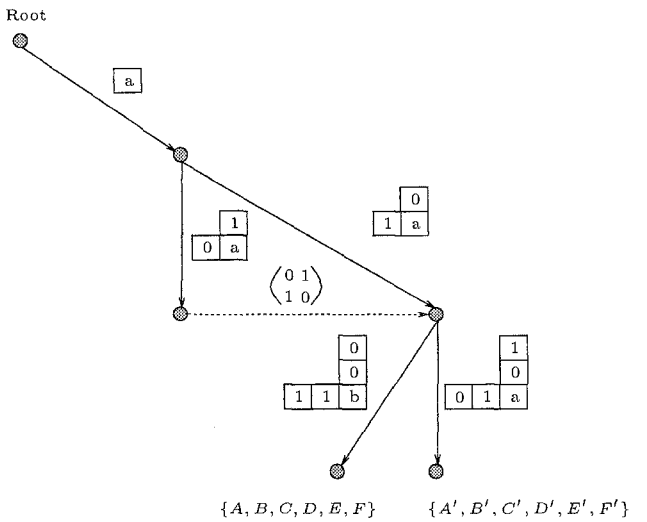


Fig. 3. Decision tree of Fig. 2 (for the graphs g_1 and g_2) after breadth-pruning was performed.

remaining vertices of G there may be an unique row-column element that must be classified. However, if we allow that the last $n-k$ columns and $n-k$ rows of the matrix M_I are rotated at run time until a classification is possible, then we can prune $n-k-1$ of the $n-k$ successor nodes of N. Breadth-pruning reduces the size of a decision tree by a factor of n. Hence, a decision tree that is subject to both redundancy- and breath-pruning contains at most $O(L3^n/n)$ nodes compared to $O(Ln^n)$ nodes in the complete decision tree. In Fig. 3, the decision tree from Fig. 2 is displayed after breadth-pruning was performed. While redundancy-pruning has already removed four nodes from the original decision tree (Fig. 1), the breadth-pruning technique succesfully removes another five nodes, namely (2,4,5,10,11). Hence, the resulting tree consists of six nodes only. The run time complexity, on the other hand, is increased by a factor of n, as there are $O(n)$ rotations necessary on each level in the decision tree. Thus, the total number of steps for graph isomorphism detection based on a breadth-pruned decision tree is $O(n^3)$. Clearly, breadth-pruned decision trees guarantuee graph isomorphism in polynomial time, independent of the number of model graphs in the database. However, due to its very nature, breadth-pruning generates decision trees which can only be used for the detection of graph isomorphisms but not the detection of subgraph isomorphisms.

5 Experimental Results

In order to test the behavior of the proposed algorithm in practice, we performed a number of experiments with randomly generated graphs. The algorithm was implemented in C++ and run on a SunSparc Workstation. For each experi-

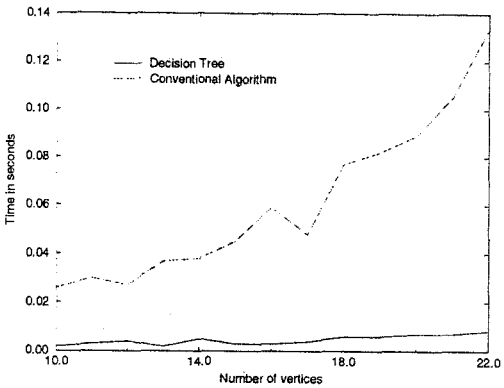


Fig. 4. First experiment.

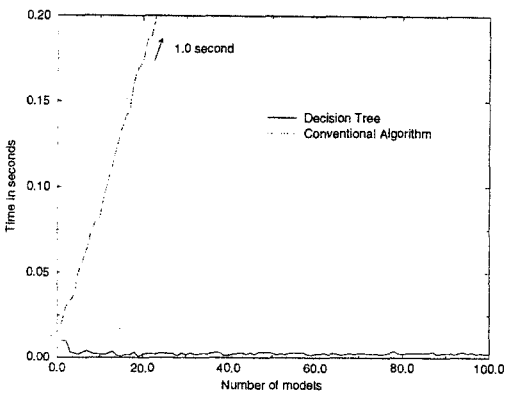


Fig. 5. Second experiment.

ment, a unlabeled model graph and an isomorphic unlabeled input graph was generated. The new algorithm was then used to detect all graph isomorphisms between the model and the input graph and the required computation time was measured. Notice that in all of the experiments, both redundancy- and breadth-pruning was applied. For comparison reasons, we ran the same experiments with a conventional algorithm (Ullman's algorithm [2]) as well. In the first experiment (Fig. 4), the size of the graphs was increased from 10 to 22 vertices and the influence on the performance was studied. Clearly, the new algorithm was faster than the conventional algorithm and its performance decreased less fast. However, the size of the decision tree grew exponentially, i.e., while there were only 174 nodes for a graph of 10 vertices there were already more than 50'000 nodes for a graph of 22 vertices. Thus, the application of the new algorithm is currently limited to graphs with no more than 22 vertices.

In the second experiment (Fig. 5), the number of model graphs in the database was steadily increased from 1 to 100 models. Each of the model graphs consisted of 11 vertices and 33 edges. As was expected from the theoretical complexity analysis, the new algorithm was completely independent of the size of the database, requiring at most 0.004 seconds for the detection of all graph iso-

morphisms from an input graph to any of the model graphs. The conventional algorithm on the other hand was linearly dependent on the number of models and required for 100 models more than 1 second.

6 Conclusions

We have presented a new algorithm for the problem of graph and subgraph isomorphism detection. While its memory requirements are exponential, it is guarantueed to run in polynomial time only. Thanks to redundancy- and breadth-pruning, graphs with up to 22 vertices can still be handled by the new algorithm. We believe that the new algorithm is an interesting option for applications which deal with large databases of moderately sized graphs and for which real-time processing is crucial.

Acknowledgment
This work is part of a project of the Priority Program SPP IF, No: 5003-34285, funded by the Swiss National Science Foundation.

References

1. H. Bunke. Structural and syntactic pattern recognition. In C.H. Chen, L.F. Pau, and P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 163–209. World Scientific Publ. Co. Singapore, 1993.
2. J.R. Ullman. An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23(1):31–42, 1976.
3. W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 17(8):749–764, 1995.
4. R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 11(11):1168–1180, 1989.
5. C.M. Hoffman. *Group-theoretic Algorithms and Graph Isomorphism.* Springer Verlag, 1982.
6. E.M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, pages 42–65, 1982.
7. B.T. Messmer and H. Bunke. Subgraph isomorphism in polynomial time. Technical Report IAM-95-003, University of Bern, 1995.

Line Postprocessing Using Probabilistic Relaxation and Relative Measurements

P. Papachristou, M. Petrou and J. Kittler

University of Surrey, Dept. of Electronic and Electrical Engineering,
Guildford GU2 5XH,United Kingdom

Abstract. In this paper, we propose a method for line extraction in complex images. We consider a two stage process. Firstly the initial measurements at each image point are given by combining the image convolution outputs obtained from directional filters. Secondly a postprocessing stage based on probabilistic relaxation refines the initial labelling using contextual information. The major difference with other similar approaches is the explicit inclusion of the relative measurements between pixels, and the careful modelling of the probability density functions of these measurements.

1 Introduction

The problem of line detection is an important task in image processing. The most common approach for line detection is that of filtering. The need to design filters for line detection different from those used for edge detection arises from the fact that the commonly used edge detectors have been designed to respond optimally to step edges and they usually detect as double edges wide linear features. A great deal of literature exists on the subject and especially on optimal filtering. See [5] for references. Petrou [5] derived a set of criteria appropriate for the characterization of filters for wide linear features and optimized them in order to define filters of various sizes.

Although a lot of effort has been put into the filtering stage of the problem, the non-linear stage of the process, namely that of post processing, has been largely neglected. In general, the output of most algorithms tends to be fragmented and noisy, requiring careful post-processing. The main drawback of many of the line detection algorithms is that the classification of a pixel as a line is made without taking into consideration the line structure in the neighbourhood of the pixel. Our approach to the problem concentrates on automatically locating the linear features as well as preserving their lateral continuity. To achieve this, we consider a two stage process which utilizes a probability relaxation labelling technique to improve the results. Firstly, an initial measurement at each image point is given by combining the image convolution outputs obtained from four directional filters. This measurement is evidence used for compiling an initial estimate of the probability of the pixel belonging to the class of line segments. Then a dictionary-based probabilistic relaxation is applied in order to refine iteratively the initial probability estimates via contextual constraints and binary relations imposed by neighbouring objects. In doing so, support is incorporated from an increasingly larger neighbourhood; the final label assignment eventually

reflects the global structure which should give a consistent and well connected line map.

This paper is organized as follows: In section 2 we summarize the filtering stage of the approach. In section 3 we summarize the results obtained by Christmas *et al.*[1] and Papachristou *et al.*[2]. The peculiarities of the probabilistic relaxation algorithm for line detection are discussed in section 4. In section 5 we apply the techniques discussed earlier to the problem of line detection. Finally, we present some experimental results in 6 and conclude in section 7.

2 The Optimal Directional Filters

Petrou [5] followed an approach similar to Canny's approach for edge detection and developed a family of convolutional filters appropriate for the detection of lines and ridges.

As the filters are model based, the algorithm Petrou [5] proposed takes full advantage of this. First of all, from the line model used, the expected response of the filter to the particular lines it is supposed to detect can be synthesized, and used as a template to decide in the first place whether a pixel is likely to be a linel or not. Then the attributes of the possible linel can be estimated as its contrast and orientation. The contrast is defined from the shape of the filter output: It consists of an extremum surrounded by two extrema of the opposite sign in symmetrical positions on either side of it. The positions of these satellite extrema depend on the width of the lines that are detected, and they can be predicted from the model response. Petrou defined as contrast of the linel the absolute difference in magnitude between the value of the filter output at the central lobe and the average value at the two satellite lobes. Although these filters are sensitive to the width of the line to be detected, they are not too sensitive to prohibit the detection of all the lines of similar width irrespective of their true orientation. (The filters respond correctly to lines that appear to them as if they had width up to 1.5 times different from the width the filters are tuned to detect). Thus, with two convolutions along two orthogonal orientations, all lines in the image with the appropriate width can be detected. The ratio of the two convolution outputs at each position reflects the local line orientation with a twofold ambiguity. This ambiguity can be resolved by performing two extra 1D convolutions along the diagonals of the image [3]. Thus, the filtering stage offers also the option to estimate the orientation of the linels with the help of a look up table.

3 Theoretical framework for object labelling using probabilistic relaxation

Christmas et al [1] and Kittler *et al.*[2] have shown that the label θ_i of an object i can take the most probable value ω_{θ_i} from a list of possible label values Ω, given a set of attributes and relations concerning the objects to be labelled, if

the labels of the objects to be labelled are updated according to the following iterative scheme:

$$P^{(n+1)}\left(\theta_i=\omega_{\theta_i}\right)=\frac{P^{(n)}\left(\theta_i=\omega_{\theta_i}\right)Q^{(n)}\left(\theta_i=\omega_{\theta_i}\right)}{\sum_{\omega_s\in\Omega}P^{(n)}\left(\theta_i=\omega_s\right)Q^{(n)}\left(\theta_i=\omega_s\right)} \tag{1}$$

where,

$$Q^{(n)}(\theta_i=\omega_{\theta_i})=\frac{1}{\hat{p}(\theta_i=\omega_{\theta_i})}\sum_{\omega_{\theta_l},l\in I_i^*}\hat{p}\left(\theta_j=\omega_{\theta_j},\forall j\in I_i\right)$$

$$\left\{\prod_{j\in I_i^*}\frac{P^{(n)}\left(\theta_j=\omega_{\theta_j}\right)p\left(\mathcal{A}_{ij}\mid\theta_i=\omega_{\theta_i},\theta_j=\omega_{\theta_j}\right)}{\hat{p}\left(\theta_j=\omega_{\theta_j}\right)}\right\} \tag{2}$$

is the support function expressing the contextual support a certain labelling receives at the nth iteration step from all other possible labels in the scene, taking into consideration the binary relations $\mathcal{A}_{ij}$ that exist between them and object i. The quantity $\hat{p}\left(\theta_j=\omega_{\theta_j},\forall j\in I_i\right)$ represents the prior joint probability of a certain label configuration of the contextual neighbourhood I_i to exist, while $\hat{p}(\theta_i=\omega_{\theta_i})$ is the prior probability of a particular label to be assigned to an object, and I_i^* is the context conveying neighbourhood of object i with the object itself excluded. The quantity $p\left(\mathcal{A}_{ij}|\theta_i=\omega_{\theta_i},\theta_j=\omega_{\theta_j}\right)$ represents the error distribution in the measurement of the binary relation $\mathcal{A}_{ij}$ given that object i has been identified with label ω_{θ_i} and object j has been identified with label ω_{θ_j}. Clearly, in the absence of noise, this probability would be a delta function at the value of the corresponding binary relation between the two labels. The iteration scheme is initialized by assigning to all objects the probabilities dictated by the object attributes only.

4 Peculiarities of the Probabilistic Relaxation algorithm for line detection

The algorithm described in section 2 for the detection of linear features offers the ground for a very interesting application of the theory of probabilistic relaxation summarized in the previous section. Although the problem is similar to that of edge detection, it has its own peculiarities that stem from the description of the filtering approach described above. For a start, the set of labels used has to be different from that used for edge detection, as one can have linear features brighter than their surroundings and linear features darker than their surroundings.

The other peculiarity of the problem is the fact that the magnitude or strength of an element in this case is not the magnitude of a vector that is the case in edge detection. A linel is endowed with the contrast and the orientation which are two scalars related in a peculiar way with each other, while an edgel is endowed with the magnitude and orientation of the *same local vector*, namely the local gradient.

5 Application of the Probabilistic Relaxation to line detection

The context conveying neighbourhood of a pixel consist of its 8 neighbours. We are going to use four labels which represent lines parallel to the two lattice axes and with positive or negative contrast and the no-line label: $\Omega = \{-\bullet-, \phi, -\circ-, \phi, \emptyset\}$. The dictionary consists of the set of 3×3 permissible linel configurations. This includes 181 permissible combinations of labels where 90 are combinations of lines with positive contrast and another 90 identical but referring to lines with negative contrast plus a configuration of all blank pixels. The *a priori* probability $\hat{p}\left(\theta_j = \omega_{\theta_j}\right)$ is determined using the appropriate configuration-frequencies from the dictionary.

5.1 The calculation of the initial probabilities

The shape of the filter output in not only distorted by noise but also by the misalignment of the direction of the convolution and the direction orthogonal to the linear feature. Thus, in order to be able to determine the probability with which a certain output has been caused by the presence of a linear feature of the specified type, we must first remove the orientation effect. We can imagine that we have a whole library of templates which correspond to the right parameters, one for each possible feature amplitude and possible orientation. At each pixel position we match all these templates in turn and choose the one that matches best. Each linear feature is usually visible by both orthogonal convolutions (unless it is very closely aligned with one of the axes) and thus, when we apply this template fitting process in each output separately, we may get inconsistent answers. To avoid this problem, we decided to treat the outputs of the two convolutions as one output, by simple concatenation: For each possible orientation of the linear feature we expect to have a certain output in the horizontal and a certain output in the vertical direction. We choose first which of these concatenated templates fits best the concatenated filter outputs by computing:

$$v_{\phi,c} = \sum_{j=-k_\phi}^{j=k_\phi} \frac{\left(R_j^0 - \mathcal{R}_j^{\phi,c}\right)^2}{\sigma^2} + \sum_{j=-k_{(90-\phi)}}^{j=k_{(90-\phi)}} \frac{\left(R_j^{90} - \mathcal{R}_j^{(90-\phi),c}\right)^2}{\sigma^2} \tag{3}$$

where variable R_j denotes the local output of the filter, $\mathcal{R}_j^{\phi,c}$ is the expected response of the convolution at position j for a feature that is oriented at ϕ with respect to the direction of convolution, and has intrinsic contrast c. We use the superscript 0 or 90 to the quantity R_j to indicate which convolution output. The length of the expected template response at orientation ϕ is $2\,k_\phi + 1$. We assume that we are dealing with additive Gaussian noise in the image with standard deviation σ_g. Then variable R_j is a Gaussianly distributed random variable with mean $\mathcal{R}_j^{\phi,c}$ and variance σ^2 given by:

$$\sigma^2 = \sigma_g^2 \sum_{i=1}^{M} f_i^2 \tag{4}$$

where f_i are the filter weights and M is the size of the filter.

Quantity $v_{\phi,c}$ is expected to follow the probability density function given by:

$$f(v) = \frac{1}{2^{n/2}\Gamma(n/2)} x^{(\frac{n}{2}-1)} e^{\frac{-v}{2}} \qquad v \geq 0 \tag{5}$$

where $\Gamma(n)$ is the *gamma function* or as it called the generalized factorial and n is the number of degrees of freedom. Experimentation showed that the best results were obtained for $n = 2(k_\phi + k_{(90-\phi)}) - 4$.

From this analysis we effectively compute the probability of the pixel under examination to be a linear feature at all. The complement of this probability with respect to 1 gives the probability of the no-line label. To distribute the probability of a line label between the two labels, the vertical and the horizontal, the following scheme was adopted:

$$P^{(0)}(\theta_i = \text{—•—}) = \begin{cases} P(\text{line}|v)\,\frac{1}{1+\tan(\theta)} & \text{if } R_i^0 \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$P^{(0)}(\theta_i = \text{•}) = \begin{cases} P(\text{line}|v)\,\frac{\tan(\theta)}{1+\tan(\theta)} & \text{if } R_i^{\frac{\pi}{2}} \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$P(\theta_i = \text{—o—}) = \begin{cases} P(\text{line}|v)\,\frac{1}{1+\tan(\theta)} & \text{if } R_i^0 \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$P(\theta_i = \phi) = \begin{cases} P(\text{line}|v)\,\frac{\tan(\theta)}{1+\tan(\theta)} & \text{if } R_i^{\frac{\pi}{2}} \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

(6)

5.2 The Binary Relations Density Functions

We consider two binary relations, namely the difference in contrast ${\mathcal{A}_{ij}}^{(1)}$ and the relative orientation between the values assigned to the two linels ${\mathcal{A}_{ij}}^{(2)}$. In order to define the probability density functions of the values of these binary relations we assume that the binary measurements associated with each pixel are independent from each other and from one pixel to the next. We will compute first the probability density functions of the contrast and orientation for each linel position, for the line and no-line cases separately and then compute the probability density functions of the relative values of these quantities.

As mentioned earlier, the contrast of a linel is measured as the difference between the filter output at the position of the linel minus the average filter output at two symmetric positions on either side of the linel. It can be shown [4]

that the probability density function of this measurement under the assumption of additive Gaussian noise to the image is well represented by

$$f_c(c) = \frac{1}{2\sqrt{\pi}\sigma} e^{-\frac{(c-\bar{c})^2}{4\sigma^2}} \tag{7}$$

where σ is given by 4 and $\bar{c}$ by

$$\bar{c} = A\left(\sum_{i=1}^{M} f_i u(f_i) - \frac{1}{2}\sum_{i=1}^{M} f_i u(-f_i)\right) \tag{8}$$

with A being the average true contrast of the lines in the image and $u(x)$ is the unit step function.

As mentioned earlier, the orientation of a linel is estimated from the ratio of the responses along two orthogonal directions with the help of a look up table. The results of the convolutions along the remaining two directions are only used in a consultative way to resolve the twofold ambiguity in orientation.

In order to estimate the probability density function of the estimated orientation (θ), we approximated by analytic expressions this look up table and the filter responses c_x and c_y along the two orthogonal directions, as functions of the intrinsic contrast c of the line (along the direction orthogonal to the line direction) and the true orientation α of the line. Using these analytic approximations, we eventually found that the probability density function of the orientation θ of a linel can be approximately represented by:

$$f_\theta(\theta) = \frac{1}{(2\pi\sigma(G^2+1)^{3/2})} \exp\left(-1/2\frac{(K_1 G - K_2)^2}{(G^2+1)\sigma^2}\right) \times \tag{9}$$

$$\left(2\sigma\sqrt{G^2+1}\exp(-F^2) + \sqrt{2\pi}(G\,K_2 + K_1)\,\mathrm{erf}(F)\right) \tag{10}$$

where

$$K_1 = c\,(1 - e^{(-7+\alpha/10)}) \qquad K_2 = c\,(1 - e^{(2+\alpha/10)})$$

$$G = \frac{\ln\left(\frac{\theta-45/2}{135/2-\theta}\right)}{s} + 1 \qquad F = \frac{G\,K_2 + K_1}{\sigma\sqrt{2(G^2+1)}}$$

Several plots of this function for different values of the actual orientation α with $c = 3$ and $s = 3$ can be seen in figure 1. (s is a parameter expressing the sharpness of the line profiles, see [5].) It is evident that the density function depends on the true orientation α. This situation is complicated and since we do not know the true orientation we will adopt as representative probability density function for the estimated orientation the one which occurs when the true orientation α is 45°.

As it can be seen from figure 1, this function has a Gaussian shape and it can be shown that it can be approximated by a Gaussian with mean the true orientation α of the linel and standard deviation $\tilde{\sigma} = 0.9 \times \frac{c}{\sigma}$, when θ is measured

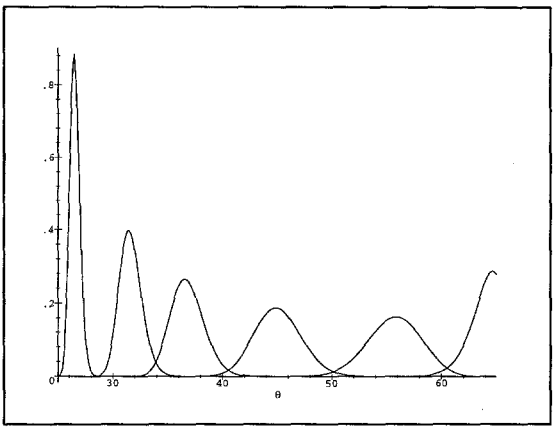


Fig. 1. The probability density function $f_\theta(\theta)$ for various values of the actual orientation $\alpha = 25, 30, 35, 45, 55, 60$ from left to right.

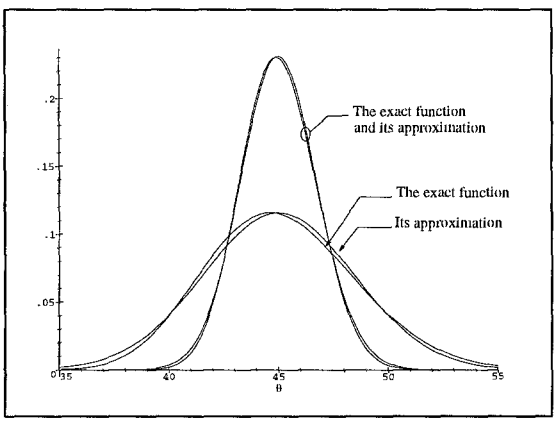


Fig. 2. The exact $f_\theta(\theta)$ function and the Gaussian approximation of it for two different signal to noise ratios c/σ.

in rads. Figure 2 shows the exact $f_\theta(\theta)$ function and the Gaussian approximation of it. As we see, the two functions are very close.

We can calculate next the probability density function of the *relative* values of the contrast and orientation between two neighbouring pixels for various label combinations and with the help of the probability density function of the measurements that refer to single pixels. We find for the various cases:

$$p\left(\mathcal{A}_{ij}^{1} = z|(\omega_{\vartheta_i}, \omega_{\vartheta_j}) \in \{(noline, noline), (line, line)\}\right) = \frac{1}{2\sqrt{2\pi}\,\sigma} e^{-\frac{z^2}{8\sigma^2}} \quad (11)$$

For the calculation of the relative contrast in the case of line/noline labels, the pixel labelled no-line is assumed to have contrast much less than the contrast of the line pixel. Then the probability density function of the relative contrasts is expected to be a function of the contrast difference between the two pixels a and will be given by: $\frac{1}{2\sqrt{2\pi}\,\sigma} e^{-\frac{(z-a)^2}{8\sigma^2}}$ Since we are not really interested in the exact value of a but for values larger than a certain threshold, we will use a sigmoid function which approximates the envelope which this family of the functions creates by joining their maxima for a greater than a threshold $a = 3\sigma$ (see figure 3). Thus, for the application we shall use:

$$p\left(\mathcal{A}_{ij}^{1} = z|(\omega_{\vartheta_i}, \omega_{\vartheta_j}) \in \{(noline, line), (line, noline)\}\right) = \frac{1}{8\sqrt{\pi}\sigma}\left[1 + erf\left(\frac{z}{2\sigma}\right)\right]$$

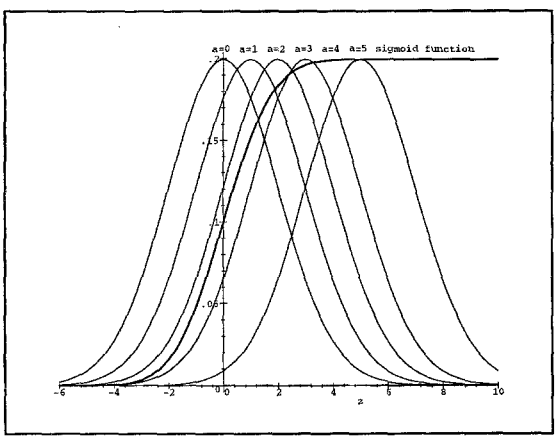


Fig. 3. The density function f_z for difference in contrast in the line/noline case (a =0,1,2,3,5 and $\sigma = 1$) and its approximation with a sigmoid function.

Finally, for the acute difference of orientation we find [4]:

$$p\left(\mathcal{A}_{ij}^{2} = \phi|(\omega_{\vartheta_i}, \omega_{\vartheta_j}) = (noline, noline)\right) = \frac{1}{\pi} \quad (12)$$

$$p\left(\mathcal{A}_{ij}^{2} = \phi|(\omega_{\vartheta_i}, \omega_{\vartheta_j}) \in \{(line, line)\}\right) = \frac{1}{2\tilde{\sigma}\sqrt{\pi}}\left(e^{-\frac{\phi^2}{4\tilde{\sigma}^2}} + e^{-\frac{(\phi-\pi)^2}{4\tilde{\sigma}^2}} + e^{-\frac{(\phi+\pi)^2}{4\tilde{\sigma}^2}}\right) \quad (13)$$

$$p\left(\mathcal{A}_{ij}^2 = \phi | (\omega_{\vartheta_i}, \omega_{\vartheta_j}) \in \{(noline, line), (line, noline)\}\right) = \frac{1}{2\,\tilde{\sigma}\,\sqrt{\pi}}\left(e^{-\frac{\phi^2}{4\tilde{\sigma}^2}} + e^{-\frac{(\phi-\pi)^2}{4\tilde{\sigma}^2}} + e^{-\frac{(\phi+\pi)^2}{4\tilde{\sigma}^2}}\right) \tag{14}$$

6 Experimental Results

Our main objective is to robustly and unambiguously extract line features from imagery containing lines, noise and edges as well. Such a problem arises when for example the aim is the identification of a road network in a remotely sensed image. Edge contamination comes from structures such as field boundaries and urban areas.

Figure 4 shows the results of applying our postprocessing algorithm to some real images. The second column of panels shows the initial probabilities as grey level values: the darker the pixel, the higher the probability to be a line. Panels in the third column are the outputs after a few iteration steps. It can be seen that gaps were filled and most of the noisy segments were removed by the postprocessing process. Since the relaxation uses an explicit dictionary model of the junction structures most of the junctions are accurately detected. Moreover, objects delineated by intensity edges are largely ignored.

7 Conclusions

We have applied here the theory of probabilistic relaxation to the problem of the detection of linear features. We have described a statistical framework which can be used to combine the responses of an optimal filter and give a probabilistic representation which is suitable for refinement by the relaxation process developed. We have demonstrated these results on synthetic and remotely sensed images.

References

1. Christmas, W., Kittler, J. and Petrou, M., "Structural matching in computer vision using probabilistic relaxation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-17**, 1995, pp. 749–764.
2. Kittler, J., Papachristou, P. and Petrou, M., "Combining evidence in dictionary based probabilistic relaxation", *8th Scandinavian Conference on Image Analysis*, **Tromso, Norway**, 1993, pp. 785–795.
3. Ng, I. and Petrou, M., "On the analysis of linear textures", in *Progress in Image Analysis and Processing II*, Villa Olmo, Como, Italy, World Scientific, 4–6 September 1991 1991, pp. 183–190.
4. Papachristou, P., *Probabilistic Relaxation for Square Lattice Structures*, PhD thesis, University of Surrey, 1994.
5. Petrou, M., "Optimal convolution filters and an algorithm for the detection of linear feature", *IEE Proc. I Communications, Speech and Vision*, 1993, pp. 331–339.

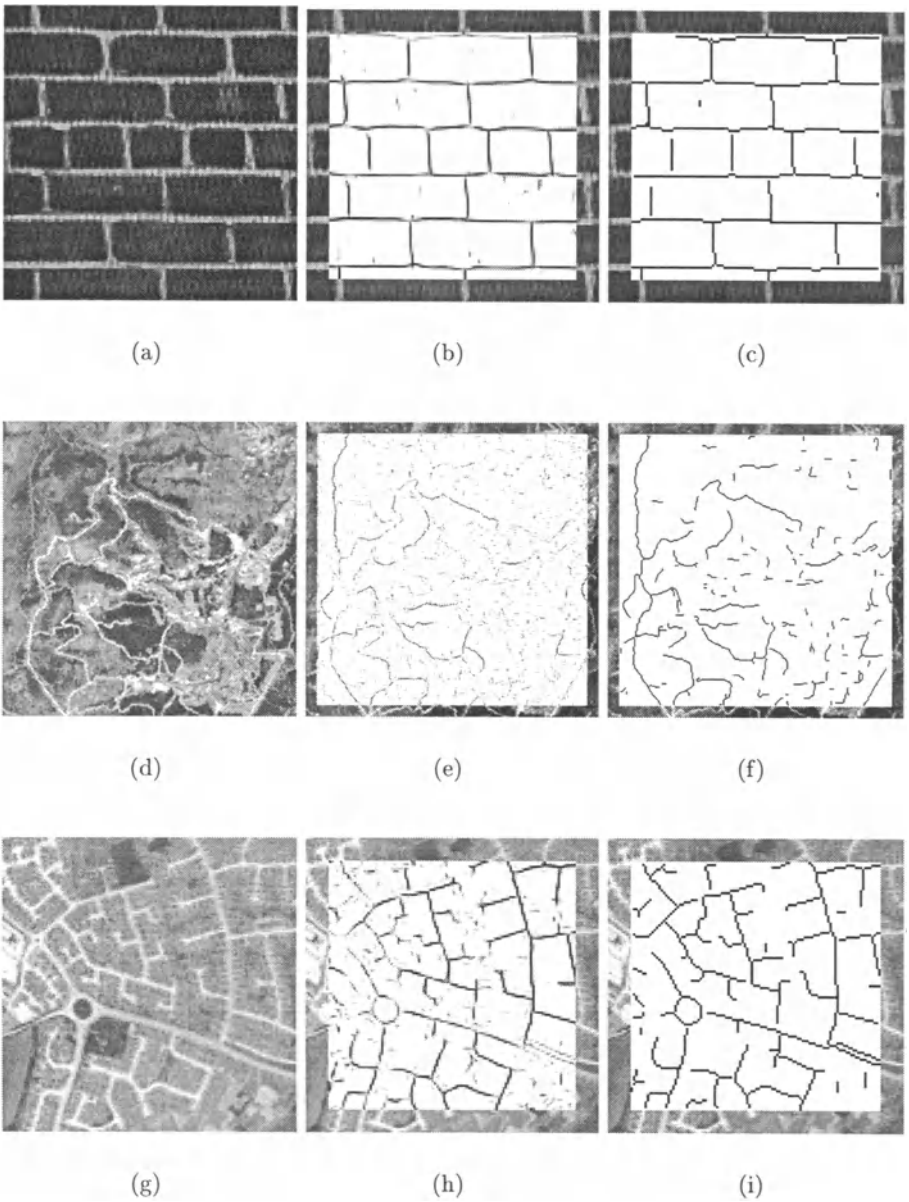


Fig. 4. First column: Original images. Second column: Initial labellings. Last column: Final labellings after 8, 11 and 13 iterations from top to bottom respectively.

Geometric Invariance

Multiple-View Geometry and Photometry

Amnon Shashua

Technion — Israel Institute of Technology
Department of Computer Science
Haifa 32000, Israel.
e-mail: shashua@cs.technion.ac.il

Abstract

The issue of how to represent and manipulate the information arising from a multitude of perspective pictures of a 3D scene is a recent and growing topic of interest. In this paper we present a summary account of up-to-date research, my own and with colleagues, on this topic, including: the trilinear constraints and associated tensor; the properties of the trilinear tensor and their relevance to camera geometry and invariance; rank deficiencies and $N > 4$ multiple-view geometry; and the photometric duality in the form of the "Tensor Brightness Constraint".

1 Introduction

The algebraic and geometric relations across multiple perspective views is a recent and growing interest which is relevant to a number of topics including (i) issues of 3D reconstruction from 2D data, (ii) representations of visual scenes from video data, (iii) image synthesis and animation, and (iv) visual recognition and indexing.

Typical to these topics is the question about the limitations and possibilities of going from two-dimensional (2D) measurements of point matches (correspondences) across two or more views to properties of the three-dimensional (3D) object or scene. Since the relationship between the 3D world and the 2D image space combines together 3D shape parameters, camera viewing parameters and 2D image measurements, the question of limitations and possibilities, in its widest scope, is about (i) 2D constraints across multiple views (matching constraints), (ii) characterizations of the space of all images of a particular object (indexing functions). In other words, one seeks to best represent, in terms of efficiency, compactness, flexibility and scope of use, two kinds of manifolds: (i) the manifold of image and viewing parameters (invariance to shape), and (ii) the manifold of image and object parameters (invariance to viewing parameters).

A further distinction of this line of research is that one generally prefers to find linear functions that describe the inherently non-linear relationship between 3D objects and their 2D views. One can argue whether for a particular application, say landscape reconstruction from aerial photographs, it really matters if the final computational method is linear or non-linear (especially when a good initial guess can be obtained for the numerical solution), but in general, seeking

ways to embed the non-linearities into a space where the manifold in question becomes a linear subspace constitutes an important step forward in understanding and manipulating visual information.

On the photometric domain an issue of interest is how to combine brightness information from images, in the form of spatial and temporal derivatives, with geometric constraints arising from the fact that all images are coming from the same 3D static scene. This issue is relevant to the problem of recovering correspondences from a sequence of images, and to the problem of recovering scene structure and observer motion parameters *directly* from brightness measurements. An important step in achieving these goals is being able to recast the general motion estimation problem into a parametric framework, i.e., that each pixel contributes measurements to a fixed number of unknowns. The latter implies a process of elimination (either structure of motion parameters), which brings us back to the geometric goals of embedding the non-linear view-manifold in spaces where it becomes a linear subspace — as we shall see in the sequel.

Finally, the meeting point between geometry and photometry comes from considering lines in addition to points. A line can be interpreted as coming from a 3D line (a geometric entity), or as an uncertainty caused by insufficient brightness measurements (the so called "aperture" problem) — a photometric entity.

We will show in this paper that common to all these goals is the so called "trilinear tensor".

1.1 Background on the Trilinear Tensor

The first work to combine points and lines into one framework is due to Aloimonos & Spetsakis [18]. They have shown that three views admit a set 27 numbers arranged in three matrices. Matching points were shown to produce three linear equations for the 27 numbers (hence 9 points are required), and matching lines produce two equations (hence 13 lines are required). They have also addressed the connection between lines and uncertainty by showing that the aperture problem which is inherent in two views does not exist with three views. This line of research continued with Shashua [13] by attempting to generalize the "linear combination" result of Ullman & Basri [20], showed that the point geometry of three views actually produces 4 linearly independent equations, each having a trilinear form in image coordinates, for a tensor of 27 numbers (which happen to be the same numbers of [18]). These equations were called "trilinearities". Thus 7 matching points across three views are sufficient for linearly generating the intrinsic geometry of three views. In that work the equation relating the tensor and the (projective) motion parameters was first derived. Hartley [5] rederived the tensor equation but with a different indexing scheme (adopted later in this paper), and Shashua & Werman [17] have shown that certain rearrangements of the tensor elements (equivalent to certain contractions of the tensor) produce *collineations* of the 2D plane, i.e., projective transformations due to intrinsic planes in space — a result which is important for unfolding the

reconstruction problem in a very simple manner and for the question of multiple (> 3) views, as described in the sequel.

The trilinearities were rederived by Faugeras & Mourrain [4] using exterior algebra. That had two distinct advantages: first, a geometric interpretation was given to the trilinear equations, second, the method of derivation was simple and general and which led later to further work by Weinshall, Werman & Shashua [21] and Carlsson [2] on "dual" tensors. Similarly, the trilinearities were rederived by Triggs [19], using Penrose tensorial notations, and Heyden [7], which together with [4, 22], established the existence of quadlinear forms (with total of 81 coefficients) across four views with the negative result that further views would not add any new constraints.

2 The Trilinear Tensor

Consider two perspective views ψ, ψ' of a 3D scene. Let P be a point in 3D projective space projecting onto matching points $p \in \psi, p' \in \psi'$ in 2D projective plane. The relationship between the 3D and 2D spaces is represented by the 3×4 matrices, $[I, 0]$, $[A, v']$, i.e.,

$$\begin{aligned} p &= [I, 0]P \\ p' &\cong [A, v']P \end{aligned} \tag{1}$$

We may adopt the convention that $p = (x, y, 1)^\top$, $p' = (x', y', 1)^\top$, and therefore $P = (x, y, 1, \rho)$. The coordinates $(x, y), (x'y')$ are matching points (with respect to some arbitrary image origin — say the geometric center of each image plane). The vector v' is the translational component of camera motion and is the view of the center of projection of the first camera in view ψ'. The matrix A is a 2D projective transformations (collineation, homography matrix) from ψ to ψ' induced by *some* plane in space (the plane $\rho = 0$). In a calibrated camera setting the plane $\rho = 0$ is the plane at infinity and A is the rotational component of camera motion and $\rho = 1/z$ where z is the depth of the point P in the first camera coordinate frame. For more details on the representation and methods for projective reconstruction see [3, 6, 12, 16, 11, 1].

Let s^l_k be the matrix,

$$s = \begin{bmatrix} -1 & 0 & x' \\ 0 & -1 & y' \end{bmatrix}$$

It can be verified by inspection that eqn. 1 can be represented by the following two equations (standard method for removing the scale in that equation):

$$\rho s^l_k v'^k + p^i s^l_k a^k_i = 0, \tag{2}$$

with the standard summation convention that an index that appears as a subscript and superscript is summed over (known as a contraction). Superscripts denote contravariant indices (representing points in the 2D plane, like v') and subscripts denote covariant indices (representing lines in the 2D plane, like the

rows of A). Thus, a_i^k is the element of the i'th row and k'th column of A, and ${v'}^k$ is the k'th element of v'. Note that we have two equations because $l = 1, 2$ is a free index.

Similarly, the camera transformation between views ψ and ψ'' is

$$p' \cong [B, v'']P.$$

Likewise, let r_j^m be the matrix,

$$r = \begin{bmatrix} -1 & 0 & x'' \\ 0 & -1 & y'' \end{bmatrix}$$

And likewise,

$$\rho r_j^m v''^j + p^i r_j^m b_i^j = 0, \tag{3}$$

Note that k and j are dummy indices (are summed over) in equations 2 and 3, respectively. We used different dummy indices because now we are about to eliminate ρ and combine the two equations together. Likewise, l, m are free indices, therefore in the combination they must be separate indices. We eliminate ρ and obtain a new equation:

$$(s_k^l {v'}^k)(p^i r_j^m b_i^j) - (r_j^m v''^j)(p^i s_k^l a_i^k) = 0,$$

and after grouping the common terms:

$$s_k^l r_j^m p^i ({v'}^k b_i^j - v''^j a_i^k) = 0,$$

and the term in parenthesis is the trilinear tensor:

$$\boxed{\alpha_i^{jk} = {v'}^k b_i^j - v''^j a_i^k. \qquad i, j, k = 1, 2, 3} \tag{4}$$

And the tensorial equations (the trilinearities) are:

$$\boxed{s_k^l r_j^m p^i \alpha_i^{jk} = 0}, \tag{5}$$

Hence, we have four trilinear equations (note that $l, m = 1, 2$). In more explicit form, these functions (referred to as "trilinearities") are:

$$\begin{aligned}
x''\alpha_i^{13}p^i - x''x'\alpha_i^{33}p^i + x'\alpha_i^{31}p^i - \alpha_i^{11}p^i &= 0, \\
y''\alpha_i^{13}p^i - y''x'\alpha_i^{33}p^i + x'\alpha_i^{32}p^i - \alpha_i^{12}p^i &= 0, \\
x''\alpha_i^{23}p^i - x''y'\alpha_i^{33}p^i + y'\alpha_i^{31}p^i - \alpha_i^{21}p^i &= 0, \\
y''\alpha_i^{23}p^i - y''y'\alpha_i^{33}p^i + y'\alpha_i^{32}p^i - \alpha_i^{22}p^i &= 0.
\end{aligned}$$

Since every corresponding triplet p, p', p'' contributes four linearly independent equations, then seven corresponding points across the three views uniquely determine (up to scale) the tensor α_i^{jk}. More details and applications can be found in [13].

It readily follows from the fact that the tensor vanishes with the contraction of two covariant vectors and one contravariant vector that a matching triplet of a point and two lines provides one linear equation for the tensor elements. Likewise, two contractions with covariant vectors leaves us with a covariant vector, thus three matching lines provide two linear equations for the tensor elements. Finally, because a point is defined by the intersection of two lines, three matching points provide four linear equations for the tensor elements (as explicitly specified above).

3 Contractions, Collineations, Fundamental Matrix

Consider the contraction $e_j \alpha_i^{jk}$ where $e = (1, 0, 0)^\top$. The result is a 3×3 matrix, denoted by E_1. Similarly, let $E_2 = \alpha_i^{2k}$, and $E_3 = \alpha_i^{3k}$. We obtain a remarkably and simple result:

The three matrices E_1, E_2, E_3 are three homography matrices $E_j : \psi \mapsto \psi'$ of three distinct and intrinsic planes.

In other words, the three matrices are *collineations* of the 2D projective plane mapping points in ψ to points in ψ' induced by three distinct planes, respectively. The orientation and locations of each plane are determined by the motion parameters B, v''. More details and proofs can be found in [17].

Since 2D collineations are the building block for projective reconstruction, this result is very important. For example, the "fundamental" matrix F between ψ and ψ' can be linearly determined from the tensor by:

$$E_j{}^\top F + F^\top E_j = 0$$

which yields 18 linear equations of rank 8 for F. Similarly, cross products between columns of two collineations provide epipolar lines which can be used to recover the epipole v'. We will also use this property of the tensor to concatenate together multiple (> 3) views, as described in the sequel.

4 Tensor Brightness Constraint

Consider all lines s' in the 2D plane that are coincident with p', i.e., $s'_k p'^k = 0$. Note that the rows of the matrix s'_k are covariant vectors (represent lines in the 2D plane), thus s' is spanned by the rows of s'_k, and in turn equation (2) still holds:

$$\rho s'_k v'^k + p^i s'_k a_i^k = 0, \tag{6}$$

for all covariant vectors s' spanned by by the two covariant vectors $(-1, 0, x')$ and $(0, -1, y')$ (representing vertical and horizontal lines, respectively).

In particular, consider the linear combination with coefficients I_x, I_y which are the components of the gradient vector ∇I measured at point (x, y) in the first view ψ. Thus,

$$s' = \begin{pmatrix} -I_x \\ -I_y \\ x' I_x + y' I_y \end{pmatrix}$$

Our next step is to remove the contribution of x', y' from s' (this is the only place in equation (6) where correspondence is required) and for that purpose we will use the "constant brightness equation" due to [8]:

$$I_x u + I_y v + I'_t = 0,$$

where $u = x - x'$, $v = y - y'$ and I'_t is the discrete temporal derivative at (x, y), i.e., $I_2(x, y) - I_1(x, y)$ where I_2 and I_1 are the image intensity values of the second and first image, respectively.

After substituting the constant brightness equation in s' we obtain:

$$s' = \begin{pmatrix} -I_x \\ -I_y \\ I'_t + x I_x + y I_y \end{pmatrix} \tag{7}$$

Thus equation (6) with the covariant vector s' given by (7) is a brightness constraint equation that relates camera motion A, v', object shape ρ and brightness information in the form of spatial and temporal derivatives. The situation so far is not very different from that of Horn & Weldon [9] with the difference that here we are using an uncalibrated (projective) model, rather than the Longuett-Higgins & Prazdny [10] model of small motion, and we use a different style of notations.

The next step is identical in concept to the way the trilinear tensor was derived above. We eliminate the contribution of object shape ρ by using the third view ψ''. The constant brightness equation between the first and third view becomes: $I_x(x - x'') + I_y(y - y'') + I''_t = 0$, and by going through exactly the same steps as before we obtain:

$$\rho s''_j v''^j + p^i s''_j b^j_i = 0, \tag{8}$$

and

$$s'' = \begin{pmatrix} -I_x \\ -I_y \\ I''_t + x I_x + y I_y \end{pmatrix} \tag{9}$$

We eliminate ρ and obtain a new equation, the "Tensor Brightness Constraint":

$$\boxed{s'_k s''_j p^i \alpha^{jk}_i = 0} \tag{10}$$

The new constraint relates the observer motion and certain measurable quantities (products of elements of s' and s'') of image spatial and temporal derivatives. Because the structure of the world does not appear in the constraint

equation we have a *parametric* constraint, which means that *every* pixel with non-vanishing spatial gradient contributes an equation with a *fixed* number of unknowns. Moreover, the constraint equation is *linear* in the elements of α_i^{jk}. Therefore, in principle, we can obtain a linear least-squares solution for the elements of α_i^{jk} just by measuring spatial and temporal derivatives. More details can be found in [15].

5 Rank 4: Properties of the Tensor Manifold

The ultimate goal is to find the means for combining together the contribution of any number of views, not only three. To this end we suggest to start with investigating the space of *all* trilinear tensors and look for rank deficiencies in that space. Any finding of that sort is extremely useful because it readily allows a statistical way of putting together many views, simply by means of factorization. The main result is that trilinear tensors across $m > 3$ views are embedded in a low dimensional linear subspace.

Consider the following arrangement: we are given views $\psi_1, \psi_2, ...\psi_{m+2}$, $m \geq 1$. For each (ordered) triplet of views there exists a unique trilinear tensor. Rather than considering all triplets of views, we consider the m triplets that contain ψ_1, ψ_2, i.e., the triplets $< \psi_1, \psi_2, \psi_i >$, $i = 3, ..., m+2$. Consider each of the tensors as a vector of 27 components (arrange the components arbitrarily, but stick with this arrangement for all tensors) and concatenate all these vectors as columns of a $27 \times m$ matrix. The question is what is the rank of this matrix when $m \geq 27$? Clearly, if the rank is smaller than 27 we obtain a line of attack on the task of putting together many views. The motivation for considering this arrangement is that a view adds only 12 parameters (up to scale). It may be the case that the redundancy of representing an additional view with 27 numbers (a column vector in the $27 \times m$ matrix), instead of 12, comes to bear only at a non-linear level — in which case it will not affect the rank of the system above. Therefore, a rank deficiency implies an important property of a collection of tensors. We can prove the following result:

All trilinear tensors live in a manifold of $\mathcal{P}^{26}$. The space of all trilinear tensors with two of the views fixed, is a 12'th dimensional linear sub-space of $\mathcal{R}^{27}$.

Therefore, the rank is 12, thus each additional view adds, linearly, only 12 parameters — as expected. An immediate consequence of this result is:

A linear combination of tensors $< \psi_1, \psi_2, \psi_i >$ and $< \psi_1, \psi_2, \psi_j >$ produces an admissible tensor $< \psi_1, \psi_2, \psi >$, for some view ψ.

The corollary is not as obvious as it may seem. All tensors live in a non-linear manifold because there are algebraic dependencies among the tensor elements. Thus, the line passing through two arbitrary points on that manifold does not

necessarily live inside the manifold. The fact that it does for the selection of points described in the corollary, is therefore, not obvious. One application, for instance, of the corollary is view synthesis and animation.

A similar result applies to the space of *all* collineations (homography matrices) between two fixed views. Given some plane in space projecting onto views ψ and ψ', the corresponding image points are mapped to each other by a collineation (homography matrix), $Ap \cong p'$ for all matching pairs p, p'. Since the homography matrix A depends on the orientation and location of the planar object, we obtain a *family* of homography matrices when we consider all possible planes.

Consider homography matrices $A_1, A_2, ..., A_k$ each as a column vector in a $9 \times k$ matrix. We ask again, what is the rank of the system? It would be convenient if it were 4, because each additional homography matrix represents a plane, and a plane is determined by 4 parameters. We can prove the following result:

The space of all homography matrices between two fixed views is embedded in a 4 dimensional linear subspace of $\mathcal{R}^9$.

We can combine the rank 4 result with the result described previously that a tensor can be contracted into three homography matrices and obtain a "rank 4" result on the space of tensors, as follows. We recall from Section 3 that the tensor α_i^{jk} can be contracted into three homography matrices, associated with three distinct planes, between ψ and ψ'. Hence, consider the same situation as before where we have views $\psi_1, \psi_2, ...\psi_{m+2}$ and consider the tensors of the triplets $< \psi_1, \psi_2, \psi_i >$, $i = 3, ..., m+2$. But now, instead of arranging each tensor as a 27 column vector, we arrange it in a 9×3 block, where each column is the homography α_i^{jk}, $j = 1, 2, 3$. We obtain a $9 \times 3m$ matrix. Its rank must be 4:

A tensor of views $< \psi_1, \psi_2, \psi_3 >$ and "third" of the tensor $< \psi_1, \psi_2, \psi_4 >$, linearly span, with 12 coefficients, all tensors $< \psi_1, \psi_2, \psi >$ (over all views ψ). Each such tensor can be recovered using 6 matching points with ψ_1 and ψ_2.

We can improve and get even a tighter result on the minimal information required to linearly span the family of tensors $< \psi_1, \psi_2, \psi >$ (for all views ψ). Assume we have the fundamental matrix F and epipole v'. It is known from the work of [11] that the matrix $[v']_x F$, where $[v']_x$ is the skew-symmetric matrix associated with vector products, is a homography matrix (linearly independent from the three homography matrices provided by the tensor). We have therefore the following result:

The tensor of views $< \psi_1, \psi_2, \psi_3 >$ and the epipolar constraint (matrix F and epipole v') together linearly span, with 12 coefficients, all other tensors $< \psi_1, \psi_2, \psi >$ (running over all views ψ). Each additional view ψ contributes

linearly 12 parameters and its tensor with ψ_1, ψ_2 can be determined linearly using 6 matching points.

More details on the material presented in this section can be found in [14].

6 summary

This paper has presented results on the goal of capturing the inter-relationship, geometrically and photometrically, across multiple perspective views. The main analysis vehicle is the "trilinear tensor" which captures in a very simple and straightforward manner the basic structures associated with this problem of research. We have not described in detail the particular applications to reconstruction, recognition and animation, but these can be found in some of the references provided in the text.

References

1. P.A. Beardsley, A. Zisserman, and D.W. Murray. Navigation using affine structure from motion. In *Proceedings of the European Conference on Computer Vision*, pages 85–96, Stockholm, Sweden, May 1994.
2. S. Carlsson. Duality of reconstruction and positioning from projective views. In *Proceedings of the workshop on Scene Representations*, Cambridge, MA., June 1995.
3. O.D. Faugeras. Stratification of three-dimensional vision: projective, affine and metric representations. *Journal of the Optical Society of America*, 12(3):465–484, 1995.
4. O.D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995.
5. R. Hartley. Lines and points in three views — a unified approach. In *Proceedings Image Understanding Workshop*, Monterey, CA, November 1994.
6. R. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1040, 1994.
7. A. Heyden. Reconstruction from image sequences by means of relative depths. In *Proceedings of the International Conference on Computer Vision*, pages 1058–1063, Cambridge, MA, June 1995.
8. B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
9. B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.
10. H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B*, 208:385–397, 1980.
11. Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. In *Proceedings of the European Conference on Computer Vision*, pages 589–599, Stockholm, Sweden, May 1994. Springer Verlag, LNCS 800.

12. A. Shashua. Projective structure from uncalibrated images: structure from motion and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):778–790, 1994.
13. A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.
14. A. Shashua and S. Avidan. The rank 4 constraint in multiple (≥ 3) view geometry. Technical report, Technion, CS Dept., October 1995.
15. A. Shashua and K.J. Hanna. The tensor brightness constraints: Direct estimation of motion revisited. Technical report, Technion, CS Dept., October 1995.
16. A. Shashua and N. Navab. Relative affine structure: Theory and application to 3D reconstruction from perspective views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 483–489, Seattle, Washington, 1994.
17. A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geometry. In *Proceedings of the International Conference on Computer Vision*, June 1995.
18. M.E. Spetsakis and J. Aloimonos. Aunified theory of structure from motion. In *Proceedings Image Understanding Workshop*, 1990.
19. B. Triggs. Matching constraints and the joint image. In *Proceedings of the International Conference on Computer Vision*, pages 338–343, Cambridge, MA, June 1995.
20. S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13:992—1006, 1991. Also in M.I.T AI Memo 1052, 1989.
21. D. Weinshall, M. Werman, and A. Shashua. Shape tensors for efficient and learnable indexing. In *Proceedings of the workshop on Scene Representations*, Cambridge, MA., June 1995.
22. M. Werman and A. Shashua. Elimination: An approach to the study of 3D-from-2D. In *Proceedings of the International Conference on Computer Vision*, June 1995.

Euclidean 3D Reconstruction from Stereo Sequences with Variable Focal Lengths

Marc Pollefeys, Luc Van Gool, Theo Moons***

Katholieke Universiteit Leuven, E.S.A.T. / MI2
Kard. Mercierlaan 94, B-3001 Leuven, BELGIUM
Marc.Pollefeys,Luc.VanGool,Theo.Moons@esat.kuleuven.ac.be

Abstract. A stereo rig can be calibrated using a calibration grid, but recent work demonstrated the possibility of auto-calibration. There remain two important limitations, however. First, the focal lengths of the cameras should remain fixed, thereby excluding zooming or focusing. Second, the stereo rig must not purely translate, which however is the most natural type of motion. This also implies that these methods collapse when the motion comes close to being a translation.
The paper extends the literature to allow changes in focal lengths (these may be independent for both cameras) and purely translational motions of the stereo rig. First, the principal points of both cameras are retrieved. Changes in focal lengths are then dealt with through weak calibration. Each position of the rig yields a projective reconstruction. The projective transformation between them allows to first retrieve affine structure which subsequently is upgraded to metric structure, following the general outline described in [12].
Rather than posing a problem to the method, rig translation allows further simplifications and is advantageous for robustness.

1 Introduction

Recently, methods to obtain the Euclidean calibration of a stereo rig have been proposed [12, 3]. These methods impose some restrictions. First, all intrinsic camera parameters are assumed fixed. This implies that e.g. the camera focal lengths are not allowed to change, and therefore precludes useful adaptations to the scene such as zooming and focusing. Second, the rig is not allowed to purely translate. Unfortunately, translation is often preferable (e.g. shortest path between points). In practice, the methods only work well if the rotational motion component is sufficiently large. In this paper the existing methods are extended to cope with changes in focal length. This will also alleviate the need for general motion (thus allowing the stereo rig to purely translate).

* IWT fellow (Flemish Institute for the Promotion of Scientific-Technological Research in Industry)

** Postdoctoraal researcher of the Belgian National Fund for Scientific Research (N.F.W.O.)

2 Camera model

The camera model used here is the pinhole model, where the image is formed under perspective projection on a photo-sensitive plane perpendicular to the optical axis. Changes in focal length move the projection center along the axis, leaving the principal point[3] unchanged. This assumption is fulfilled to a sufficient degree with the cameras commonly used[7] and especially with those used in the experiments reported here. The relation between image points and world points is given by

$$\lambda_{ijs} m_{ijs} = \mathbf{P}_{js} M_i \tag{1}$$

with $\mathbf{P}_{js}$ the 3×4 camera matrix for the jth view, s stands for *left* or *right*, m_{ijs} and M_i are column vectors containing the homogeneous coordinates of the image points and world points resp., and λ_{ijs} expresses the equivalence up to a scale factor. If $\mathbf{P}_{js}$ represents a Euclidean camera, it can be put in the form [6]

$$\mathbf{P}_{js} = \mathbf{K}_{js} \left[\mathbf{R}_{js} | - \mathbf{R}_{js} t_{js}\right] \tag{2}$$

where $\mathbf{R}_{js}$ and t_{js} represent the Euclidean orientation and position of the camera with respect to a world frame, and $\mathbf{K}_{js}$ is the calibration matrix of the jth camera:

$$\mathbf{K}_{js} = \begin{bmatrix} r_{xs}^{-1} & -r_{xs}^{-1} \cos\theta_s & f_{js}^{-1} u_{xs} \\ 0 & r_{ys}^{-1} & f_{js}^{-1} u_{ys} \\ 0 & 0 & f_{js}^{-1} \end{bmatrix} . \tag{3}$$

In this equation r_{xs} and r_{ys} represent the pixel width and height, θ_s is the angle between the image axes, u_{xs} and u_{ys} are the coordinates of the principal point, and f_{js} is the focal length. Notice that the calibration matrix is only defined up to scale. In order to highlight the effect of changing the focal length the calibration matrix $\mathbf{K}_{js}$ will be decomposed in two parts:

$$\mathbf{K}_{js} = \begin{bmatrix} 1 & 0 & (f_{1s}/f_{js} - 1) u_{xs} \\ 0 & 1 & (f_{1s}/f_{js} - 1) u_{ys} \\ 0 & 0 & f_{1s}/f_{js} \end{bmatrix} \mathbf{K}_{1s} . \tag{4}$$

The second part $\mathbf{K}_{1s}$ is equal to the calibration matrix of the sth camera for view 1, whereas the first part, which will be called $\mathbf{K}_{f_{js}s}$ in the remainder of this text, models the effect of changes in focal length. From equation (4) it follows that once the principal point u_s is known, $\mathbf{K}_{f_{js}s}$ is known for any given value of f_{js}/f_{1s}. Therefore, finding the principal point is the first step of the reconstruction method. Then, if the change in focal length between two views can be retrieved, its effect is canceled by multiplying the image coordinates to the left by $\mathbf{K}_{f_{js}s}^{-1}$.

Retrieving the principal points u_s is relatively easy for cameras equipped with a zoom. Upon changing the focal length (without moving the camera or

[3] The principal point is defined as the intersection point of the optical axis and the image plane.

the scene), each image point will – according to the pinhole model – move on a line passing through the principal point. By taking two or more images with a different focal length and by fitting lines through the corresponding points, the principal point can be retrieved as the common intersection of all these lines. This is illustrated in Figure 1. In practice the lines will not intersect precisely and a least squares approximation is used, as in [7].

For the sake of simplicity, we will assume that $\mathbf{R}_{1s} = \mathbf{I}$ and $t_{1s} = 0$. This is not a restriction because the reconstruction is up to a similarity (i.e. Euclidean + scaling) anyway. In this way the 6 degrees of freedom of a Euclidean transformation are fixed. Choosing a value for the focal length f_{1s} fixes the last free parameter.

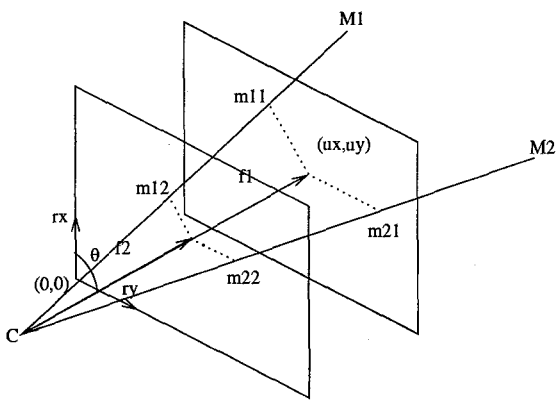


Fig. 1. *Illustration of the camera and zoom model. The focal lengths f_1 and f_2 are different, the other parameters ($r_x, r_y, u_x, u_y, \theta$) are identical.*

3 Retrieving focal length

As mentioned before, the first step in the calibration process is the recovering of the changes in the focal length for both cameras. This will be done by looking at the displacements of the epipoles. The epipoles are two points associated with a pair of cameras. The epipole in one camera image is the projection of the other camera's center. Note that the epipoles of a fixed stereo rig stay put, independent of the rig's motion. If the focal lengths of the cameras change, however, then the epipoles will shift (Fig. 2). These shifts suffice to derive the relative change in focal length. It follows from the equations for the epipoles

$$\lambda_{e_{js}} e_{js} = \lambda_{e_s}(e_s + (f_{1s}/f_{js} - 1)u_s) \tag{5}$$

that f_{js}/f_{1s} can be recovered in a linear way. In fact, the explicit calculation of focal lengths themselves is not called for. Indeed, one can transform the images to

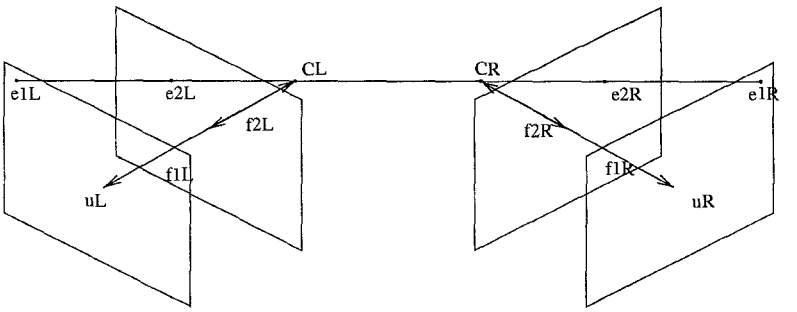


Fig. 2. *Illustration of the displacement of the epipole for the left and right camera when changing the focal length. C_L is the center of the left camera, e_{0L} and e_{1L} the epipoles in the left camera for different focal lengths (f_{0L} and f_{1L}) and u_L is the principal point. Similar notations are used for the right camera.*

what they would have been like without the change in focal lengths by applying the transformation $\mathbf{K}^{-1}_{f_j s s}$ to the image (see equation (4)).

4 Affine and Euclidean calibration

When the images are corrected as to undo the changes in focal length, calibration is obtained by generalizing the elegant method proposed by Zisserman *et al* [12]. This method first retrieves the affine calibration based on the eigenvector structure of the transformation $\mathbf{T}$ between the projective reconstructions from the two positions of the rig. Once the infinite homography is known, one can use the constraints on the camera calibration matrix described in [6, 8]. In the case of a translating stereo rig this is not enough, because the problem is ill-conditioned for any movement close to translation [12]. Hence, one has to strike a balance between the ease of setting up the system (the less calibration the better) and the flexibility it has to offer in its use (e.g. being able to perform any kind of motion and to dynamically zoom and focus). Hence, this paper gives in with respect to completely calibration-free operation in two respects. First, the principal points are extracted, which is not too difficult through the very application of changes in the focal lengths. Second, the camera axes are supposed to be orthogonal. This assumption hardly poses any restriction with CCD cameras.

4.1 Affine calibration

One view with a stereo rig suffices to get a projective reconstruction [5, 4]. Having two views yields two reconstructions, M_{iP1} and M_{iP2} say (these are 4-vectors of homogenous coordinates for the ith scene point). These two reconstructions are related by a projective transformation $\mathbf{T}_{12}$:

$$\lambda_{i12} M_{iP1} = \mathbf{T}_{12} M_{iP2} \ . \tag{6}$$

If one uses the same camera matrices for both reconstructions (i.e. $\mathbf{P}_L = [\mathbf{I}|0]$ and $\mathbf{P}_R = [[e_R]_\times \mathbf{F}|e_R]$ with $\mathbf{F}$ the fundamental matrix and e_R the epipole of the right camera[11]), $\mathbf{T}_{12}$ can be written as

$$\mathbf{T}_{12} = \mathbf{T}_{PE}^{-1}\mathbf{T}_{12E}\mathbf{T}_{PE} \ , \tag{7}$$

a conjugation of the Euclidean transformation $\mathbf{T}_{12E}$ which models the motion of the rig between the two views, with the projective transformation $\mathbf{T}_{PE}$ between the reconstructions and the Euclidean world. This observation is key to the following analysis proposed in [12]. The eigenvectors of $\mathbf{T}_{12}$ are related to these of $\mathbf{T}_{12E}$ by the same transformation $\mathbf{T}_{PE}$. Because the eigenvectors of $\mathbf{T}_{12E}$ are all in the plane at infinity π_∞, the eigenvectors of $\mathbf{T}_{12}$ indicate the position of π_∞ in the projective frame. Once π_∞ is known it is easy to get the infinity homography $\mathbf{H}_{\infty LR}$. This homography transforms the projection of the points at infinity from the left image onto their equivalents in the right image, and hence can be retrieved from at least 4 point correspondences. In general 3 eigenvectors of $\mathbf{T}_{12E}$ are independent and thus –by projection in the images– give rise to 3 correspondences. Adding the correspondence between the epipoles which are both the projection of all points on the line passing through both camera centers (including the point at infinity), allows to calculate $\mathbf{H}_{\infty LR}$. To obtain an affine reconstruction one can then use the following camera matrices[8]

$$\mathbf{P}_{LA} = [\,\mathbf{I}\,|0] \text{ and } \mathbf{P}_{RA} = [\mathbf{H}_{\infty LR}|e_R] \ . \tag{8}$$

4.2 Affine calibration for translation

If the motion of the stereo rig is restricted to a translation, there is an easier and more robust method to recover the affine structure of the scene, [9], which will now be generalized to changing focal lengths.

In case of a camera translation between two views (without changing the focal length), the epipolar geometry is the same for both images. This means that the epipolar geometry between two views obtained by the same camera is completely determined by knowing the position of the unique epipole. Adding changes in focal length between the images adds one degree of freedom when the principal point is known.

Given three points in the two views, one knows that a scaling (with respect to the principal point and equal to the focal length ratio) should bring them in a position such that the lines through corresponding points intersect in the epipole. This yields a quadratic equation in the focal length ratio. The epipole itself follows as the resulting intersection. In practice, the data will be noisy, and it is better to consider information from more points. The following equation describes the relation between the image coordinates in both images

$$\lambda_{i2s} m_{i2s} = \lambda_{i1s}\,(m_{i1s} + (f_{1s}/f_{2s} - 1)u_s) + \lambda_{e_{21s}} e_{21s} \tag{9}$$

where m_{i1s}, m_{i2s}, u_s and e_{21s} are column vectors of the form $[x\,y\,1]^\top$. Equation (9) gives 3 constraints for every point and was used to form an overdetermined

system, yielding among other things f_{2s}/f_{1s} and λ_{i1s}. This leads to a system of nonlinear equations[4], which can be solved robustly (see [10] for more details).

At this stage the affine reconstruction is trivial to obtain. From equation (1) it followse that $[\lambda_{i1s} m_{i1s}^{\top}\ 1]^{\top}$ is related to M_i by an affine transformation.

In the next section the infinity homographies will be needed. Observe that for translational motions $\mathbf{H}_{\infty 12L}$ and $\mathbf{H}_{\infty 12R}$ will be equal to $\mathbf{K}_{f_{2L}L}$ and $\mathbf{K}_{f_{2R}R}$ respectively. $\mathbf{H}_{\infty 1LR}$ can be extracted as the 3×3 upper-left submatrix of the affine transformation relating the affine reconstructions obtained from the left and the right camera respectively[5].

4.3 Euclidean calibration

To upgrade the reconstruction to Euclidean structure, the camera calibration matrix $\mathbf{K}_{1L}$ (or $\mathbf{K}_{1R}$) has to be known. This is equivalent to knowing the image $\mathbf{B}_{1L}$ of the dual of the absolute conic for the left camera, since $\mathbf{B}_{1L} = \mathbf{K}_{1L}\mathbf{K}_{1L}^{\top}$. The matrices $\mathbf{B}_{1L}$ and $\mathbf{B}_{1R}$ are constrained in the following way [6, 8, 12]:

$$\kappa_{1LR}\mathbf{B}_{1R} = \mathbf{H}_{\infty 1LR}\mathbf{B}_{1L}\mathbf{H}_{\infty 1LR}^{\top} \tag{10}$$

and for each camera:

$$\kappa_{12L}\mathbf{B}_{2L} = \mathbf{H}_{\infty 12L}\mathbf{B}_{1L}\mathbf{H}_{\infty 12L}^{\top} \tag{11}$$

$$\kappa_{12R}\mathbf{B}_{2R} = \mathbf{H}_{\infty 12R}\mathbf{B}_{1R}\mathbf{H}_{\infty 12R}^{\top} \tag{12}$$

Eqs. (11) and (12) are easier to use because κ_{12L} and κ_{12R} can be forced to 1 by taking $\det \mathbf{H}_{\infty 12L} = 1$ which gives a set of linear equations. The problem with pure translation is that eqs. (11) and (12) become trivial.

The knowledge of u_L and u_R and the orthogonality constraint is called to the rescue. Take a closer look at $\mathbf{B}_{1L}$ (or $\mathbf{B}_{1R}$ for that matter):

$$\mathbf{B}_{1L} = \begin{bmatrix} f_{1L}^2 r_{Lx}^{-2} + u_{Lx}^2 & u_{Lx}u_{Ly} & u_{Lx} \\ u_{Lx}u_{Ly} & f_{1L}^2 r_{Ly}^{-2} + u_{Ly}^2 & u_{Ly} \\ u_{Lx} & u_{Ly} & 1 \end{bmatrix} \tag{13}$$

Combining Eqs. (10) and (13) gives 6 linear equations in 5 unknowns:

$$\kappa_{LR}(f_R^2 r_{Rx}^{-2} + u_{Rx}^2) = a_{11}r_{Lx}^{-2} + b_{11}r_{Ly}^{-2} + c_{11} \tag{14}$$

$$\kappa_{LR}u_{Rx}u_{Ry} = a_{12}r_{Lx}^{-2} + b_{12}r_{Ly}^{-2} + c_{12} \tag{15}$$

$$\kappa_{LR}(f_R^2 r_{Ry}^{-2} + u_{Ry}^2) = a_{22}r_{Lx}^{-2} + b_{22}r_{Ly}^{-2} + c_{22} \tag{16}$$

$$\kappa_{LR}u_{Rx} = a_{13}r_{Lx}^{-2} + b_{13}r_{Ly}^{-2} + c_{13} \tag{17}$$

$$\kappa_{LR}u_{Ry} = a_{23}r_{Lx}^{-2} + b_{23}r_{Ly}^{-2} + c_{23} \tag{18}$$

$$\kappa_{LR} = a_{33}r_{Lx}^{-2} + b_{33}r_{Ly}^{-2} + c_{33} \tag{19}$$

[4] Notice that, in contrast to Eq.(5), we can not see λ_{i1s} and $\lambda_{i1s}(f_{1s}/f_{2s} - 1)$ as independent unknowns because $(f_{1s}/f_{2s} - 1)$ is unique for all points.

[5] These reconstructions must be built with camera centered reference frames.

where a_{ij}, b_{ij} and c_{ij} only depend on $\mathbf{H}_{\infty RL}$ and u_L (which are both known). These set of equation can be solved linearly by seeing $r_{Lx}^{-2}, r_{Ly}^{-2}, \kappa_{LR}, \kappa_{LR}(f_R^2 r_{Rx}^{-2} + u_{Rx}^2)$ and $\kappa_{LR}(f_R^2 r_{Ry}^{-2} + u_{Ry}^2)$ as the unknowns. Notice that one could just solve Eqs.(15), (17), (18) and (19). The Euclidean calibration of the left camera suffices to obtain a Euclidean reconstruction. We can upgrade the affine reconstruction (obtained by the methods described in the previous sections) to Euclidean by applying the transformation

$$\mathbf{T}_{AE} = \begin{bmatrix} \mathbf{K}_{1L}^{-1} & 0 \\ 0 & 1 \end{bmatrix} . \tag{20}$$

5 Results

The algorithm described in the previous section, was applied to synthetic images as well as real images. From tests with synthetic data one can conclude that restricting the motion to translation gives more stable results. For a report on these results, we refer to [10].

Next, some results obtained from a real scene are presented. The scene consists of a box and a cylindrical object on a textured background. Images were acquired with a translating stereo rig. They can be seen in Figure 3. Figure 4

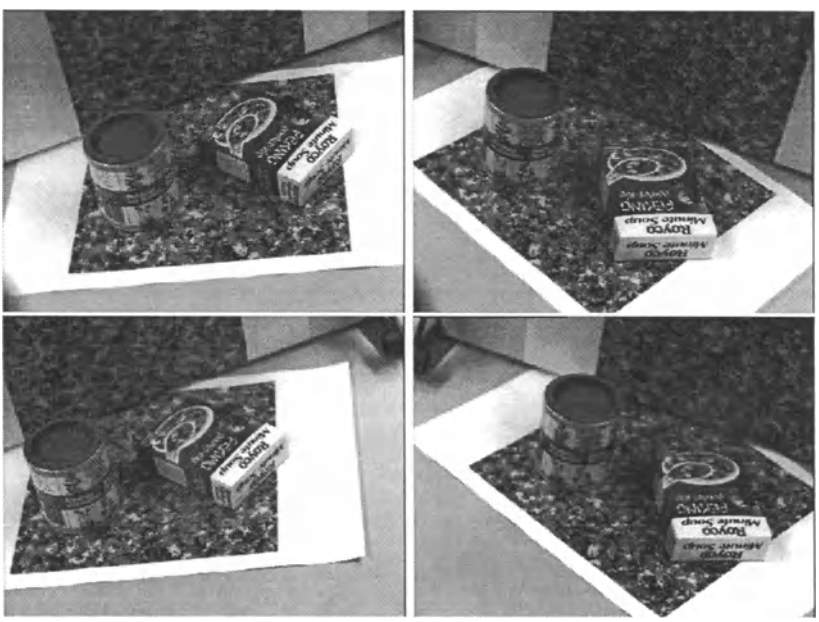


Fig. 3. *Two pairs of images of a scene taken with a translating stereo rig.*

shows the reconstruction results. Notice that angles are well preserved (e.g. the

top and the front view differ by 90°, the box and the floor have right angles in the reconstruction). The inaccuracies in the reconstruction (like the dent in the cylindrical object) are mainly due to the rendering process which uses triangulation between matched points and are not related to the accuracy of the calibration.

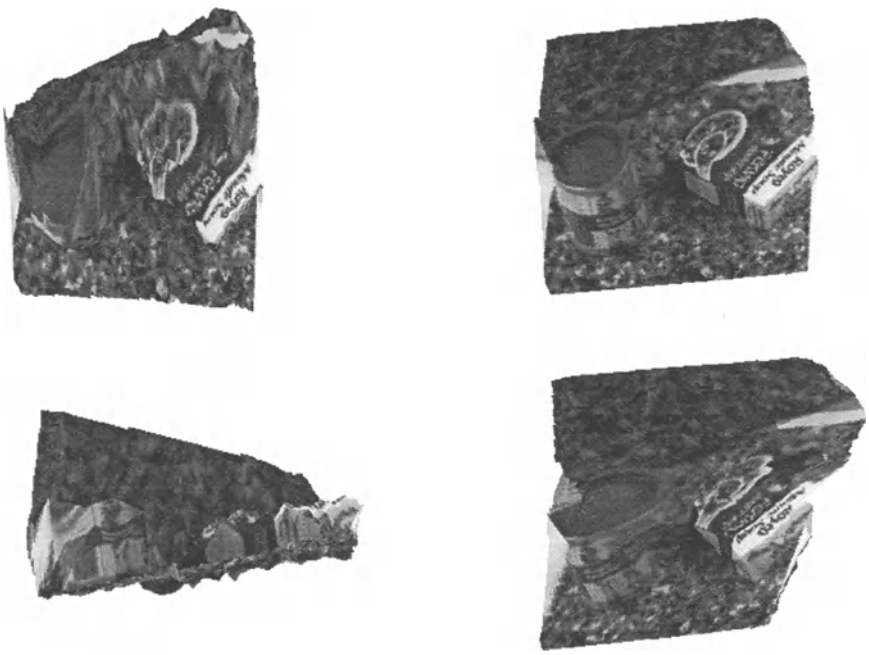


Fig. 4. *Different views of the 3D reconstruction of the scene.* Top left: *top view,* bottom left: *front view,*top and bottom right: *general views.*

6 Conclusion

The possibility to auto-calibrate a moving stereo rig with variable focal lengths is demonstrated. Only very mild forms of camera calibration had to be introduced in return. Moreover, it is shown that the method generalizes to cases of pure translation. This not only extends the existing methods, but more importantly, can also be implemented with increased robustness. The method is illustrated with a real scene. The results are convincing with respect to the Euclidean auto-calibration aspects.

Further work includes the integration of the methods into an implementation that detects the degenerated cases (i.e. translation) by itself. Also the application

of more robust techniques for the recovery of the projective structure is under investigation. Extension to variations in other parameters than focal length will be considered too.

Acknowledgement

Marc Pollefeys acknowledges a specialisation grant from the Flemish Institute for Scientific Research in Industry (IWT) and Theo Moons acknowledges a postdoctoral research grant from the Belgian National Fund for Scientific Research (N.F.W.O.). Financial support from the EU ACTS project AC074 'VANGUARD' is also gratefully acknowledged.

References

1. M. Armstrong, A. Zisserman, and P. Beardsley, Euclidean structure from uncalibrated images, *Proc. 5th BMVC*, 1994.
2. R. Deriche, Z. Zhang, Q.-T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. *Proc.ECCV'94*, pp. 567–576, Springer-Verlag, 1994.
3. F. Devernay and O. Faugeras, From Projective to Euclidean Reconstruction, *INSIGHT meeting Leuven*, 1995.
4. O. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig, *Proc.ECCV'92*, pp.321-334, 1992.
5. R. Hartley, Estimation of relative camera positions for uncalibrated cameras, *Proc.ECCV'92*, pp.579-587, 1992.
6. R. Hartley, Euclidean reconstruction from uncalibrated views, in: J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of invariance in Computer Vision*, Lecture Notes in Computer Science **825**, pp. 237–256, Springer, 1994.
7. M. Li, Camera Calibration of a Head-Eye System for Active Vision, *Proc. ECCV'94*, pp. 543–554, Springer-Verlag, 1994.
8. Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. *Proc. ECCV'94*, pp. 589–597. Springer-Verlag, 1994.
9. T. Moons, L. Van Gool, M. Van Diest, and E. Pauwels, Affine reconstruction from perspective image pairs, in : J.L. Mundy, A. Zisserman, and D. Forsyth (eds.), *Applications of Invariance in Computer Vision*, Lecture Notes in Computer Science **825**, pp. 297–316, Springer, 1994.
10. M. Pollefeys, L. Van Gool, and M. Proesmans, Euclidean 3D reconstruction from image sequences with variable focal lengths, Technical Report K.U.Leuven, E.S.A.T./MI2, 1995.
11. C. Rothwell, G. Csurka, and O.D. Faugeras, A comparison of projective reconstruction methods for pairs of views, *Proc. ICCV'95*,pp. 932-937, 1995.
12. A. Zisserman, P.A.Beardsley, and I.D. Reid, Metric calibration of a stereo rig. In *Proc. Workshop on Visual Scene Representation*, Boston, MA, June 1995.

Characterizing the Stability of 3D Invariants Derived from 3D Translational Symmetry

J. Liu[1], E. Walker[1] and J. Mundy[2]

[1] Rensselaer Polytechnic Institute Troy, New York 12180
[2] GE Corporate Research and Development Schenectady, NY 12309

Abstract

Translational symmetry provides a powerful constraint that enables recognition of three-dimensional (3D) objects from a single view. Under ideal conditions, affine invariants of 3D points reconstructed from corresponding points would serve as accurate indices for object recognition. In real images, however, inaccurate 2D coordinates of corresponding points can lead to erroneous invariants. In this paper, we investigate the effects of such errors on the 3D invariants, and discuss some approaches to successful indexing in the presence of these errors.

1 Introduction

Symmetry can provide powerful constraints to enable the recognition of three-dimensional objects from a single view. There has been considerable exploration of the properties of rotational symmetry[4] and bi-lateral symmetry [6]. In this paper, we consider the case of translational symmetry. The definition of translational symmetry is illustrated in Figure 1.

The geometric configuration of translational symmetry is composed of an object and a translated copy of the object. Under this constraint, the vectors joining corresponding points on the object are parallel. Corresponding tangent planes at any point on the object and its copy are also parallel. These two relations can be exploited to provide constraints for image feature grouping and the discovery of image locations of corresponding points on the object and its copy.

It is assumed that the image is uncalibrated, i.e., the focal length, image scale, camera orientation and viewpoint are all unknown. Over the last few years, much has been learned about reconstruction of 3D geometry from uncalibrated image views. Hartley[2] and Faugeras[1] have shown that an object can be reconstructed up to an unknown 3D projective transformation from two arbitrary uncalibrated views. In the case of views taken under simple translation, the reconstruction produces only an affine ambiguity. The translational symmetry constraint enables reconstruction of an affine description of the 3D object using standard epipolar-based triangulation, briefly reviewed in Section 3.

The reconstructed 3D points can be used to define an affine coordinate frame. Object point coordinates in this frame are invariant to affine transformations.

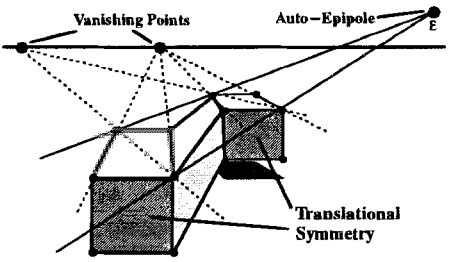


Fig. 1. An illustration of 3D translational symmetry.

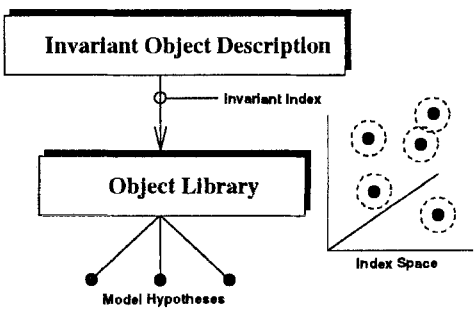


Fig. 2. Object recognition is based on the concept of image-derived invariant indices that permit efficient access to a object library.

Since the ambiguity in points reconstructed from translated objects using epipolar geometry is affine, these coordinates can serve as invariant indices for object recognition. The invariant coordinates are mapped onto a hash key that accesses the library to retrieve model hypotheses for subsequent verification.

The key requirement for effective indexing is that the variance in the index over viewpoint and image conditions is small compared to the separation of objects in index space as shown in Figure 2. The separation of the objects in index space is intrinsic to the object geometry and depends on the selection of objects in the library. The variance of the index primarily depends on errors in image feature location.

In this paper, we investigate the effect of image error on the computation of the affine invariants derived from single images of translational symmetry. The study is based on simulation of typical imaging conditions by constructing random perturbations of ideal projected image locations. Thus, the results are empirical in that numerical error bounds are found for a range of image viewpoints and orientations. These results are used to investigate a number of critical issues in the design of library index functions.

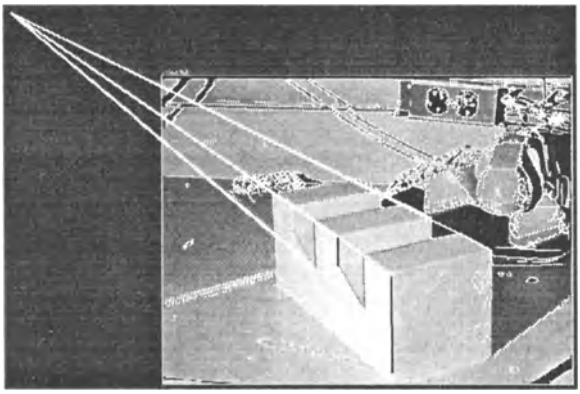


Fig. 3. Corresponding points can be obtained by grouping, using the constraints imposed by translational symmetry.

2 Grouping and Finding Correspondences

Corresponding points of translational symmetry are found by grouping image features according to the following quasi-invariant and invariant relations:

- corresponding lines should be approximately parallel;
- corresponding vertices should have incident lines which are approximately parallel;
- corresponding vertices define lines which all intersect at the epipole.

The search for correspondences starts by finding reasonably long lines which are nearly parallel. Then a chain of line segments starting at the initial line, and the corresponding chain(hypothesized) on the copy is checked for parallelism of corresponding line segments. If a sufficient number of such line segments are approximately parallel, then the epipole constraint is tested. This search continues until five or more points, with the same epipole, are obtained to compute invariants. An example of such grouping and the resulting epipole is shown in Figure 3.

3 3D Reconstruction and Invariants

This section describes in more detail the methodology for constructing invariant indices based on translational symmetry. Given the epipolar geometry of a repeated structure with a translation, 3D points are reconstructed from correspondences, and 3D affine invariants are then computed from the reconstructed points.

3.1 Epipolar Geometry

A set of parallel lines in space, under perspective, projects as a coincident bundle of lines in the image, where the common point of intersection corresponds to

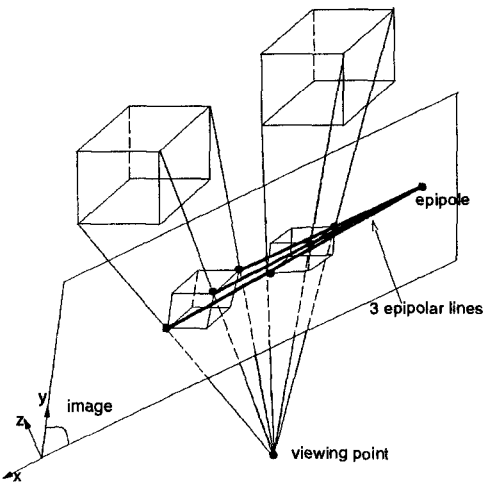


Fig. 4. 2D epipolar geometry.

a point at infinity, or vanishing point, as shown in Figure 1. In the case of translational symmetry, since all lines passing through pairs of corresponding points are parallel, a single vanishing point can be recovered from these lines. (See Figure 4). This vanishing point is called an *epipole*, and is found at $(t_x/t_z, t_y/t_z)$ [5], where (t_x, t_y, t_z) is the translation vector between the repeated structures.

3.2 Reconstruction of 3D Points

The 3D coordinates of points on the object are reconstructed using the epipolar constraints. The points in an image due to an object and its copy are given by,

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{P}$$

$$\lambda' \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} X + t_x \\ Y + t_y \\ Z + t_z \end{bmatrix} = \mathbf{P}'$$

The geometry of this configuration for one image dimension, say $\mathbf{u}$ is shown in Figure 5. Note that the epipole is the intersection of a line parallel to the translation vector, and the image plane. From this figure it is clear that:

$$\frac{t_x}{t_z} = \frac{\epsilon_u}{f}$$

where ϵ_u is the location of the epipole and $\mathbf{t}$ is the translation vector. Since we are allowing an unknown affine transformation of space we can assume a unit

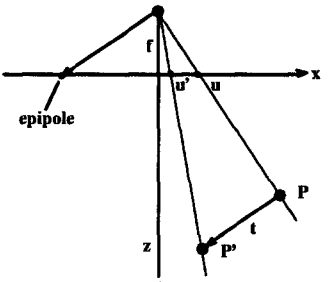


Fig. 5. The 3D reconstruction of a point on a repeated structure.

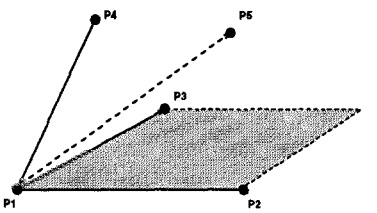


Fig. 6. An invariant affine coordinate frame. Four points define a coordinate system in which the coordinates of any fifth point are invariant to 3D affine transformations.

focal length. Solving the above equations for x and z,

$$z = t_z \frac{(u' - \epsilon_u)}{u - u'}$$
$$x = uz$$

Note that as the epipole recedes to infinity, t_z approaches zero and z reduces to the standard epipolar-aligned reciprocal disparity result. This result generalizes to two dimensions as:

$$z = t_z \sqrt{\frac{(u' - \epsilon_u)^2 + (v' - \epsilon_v)^2}{(u - u')^2 + (v - v')^2}}$$
$$x = uz$$
$$y = vz$$

3.3 Affine Invariants of 3D Points

Any four reconstructed points can be used to construct an affine coordinate frame, as shown in Figure 6. Relative to this affine coordinate frame, the coordinates of any fifth point remain invariant to a 3D affine transformation of space. Therefore, the recovered affine coordinates from any set of points reconstructed using epipolar geometry (and from any pair of translationally-related views) will be the same. These coordinates provide the invariant indices for object recognition.

The stability of these invariant indices is directly related to the error in reconstruction of the 3D points. In the remainder of the paper we investigate the effect of image feature location error on 3D reconstruction error and in turn the value of the affine invariants.

4 Experiments

The goal of our experiments is to understand how differences in camera settings and errors in feature extraction and grouping, will quantitatively affect 3D invariants. We simulate the effect of such errors by applying random perturbations to the ideal projected image locations for a known object and camera.

4.1 Experimental Setup

Since our interest is in the error introduced by reconstruction and computation of affine invariants, we assume that point correspondences points are correct. In our experiments, we simulate a SONY3000 camera, and consider the effect of point position errors of up to two pixels under typical table-top viewing conditions.

The simulation uses a cube of size 100mm, and a copy translated along one of the axes of the cube by 200mm. Specifically, the pair of cubes is created by initializing two 100mm cubes centered at coordinates (-100, 0, 0) and (100, 0, 0), and oriented parallel to the three major axes. Different viewpoints are created by rotating the cubes together, as if they were connected with a rigid bar through their centers. The center of this bar is at the origin of the coordinate system. Each viewpoint is described by the three euler angles $(\theta_x, \theta_y, \theta_z)$.

To investigate the effect of camera settings, we consider four settings of the simulated camera: 0.5, 1.0, 1.5, and 5.0 meters from the blocks. The projected object size at zero rotation is fixed by adjusting the focal lengths to 10mm, 20mm, 30mm, and 100mm, respectively. Grouping error is simulated by perturbing each projected point by a randomly chosen value from the set {-2,0,2} pixels in each of its two dimensions, based on a uniform distribution of values from the set. Using the Monte Carlo method, for each cube orientation, for each camera position, we generated 10,000 random images, and computed both the 3D reconstructions and the affine invariants for each. The value, 2 pixels, is based on typical location errors derived from line fitting and image junction intersection.

4.2 Reconstruction Errors

It is easy to show that the expected reconstructed depth error will be proportional to the square of the depth and inversely proportional to the image length of the projected translation vector. In our experiments, the error does not increase as the square of the depth, since we adjust the focal length to keep the projection of the object the same size in each case. This normalization is in accordance with typical camera setup. The error is thus proportional to depth.

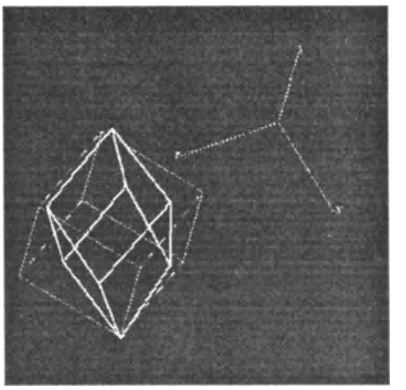


Fig. 7. Reconstruction of a cube, distance = 1.5m, $(\theta_x, \theta_y, \theta_z) = (0,60,0)$ degrees.

In the 3D reconstruction experiment, we reconstructed one corner of the cube for each of the 10,000 images, and computed the mean and standard deviation of each coordinate of the reconstructed points. As expected, the error in z was larger than either of the other two errors. Thus, reconstructed cubes tended to look distorted along the viewing direction as in Figure 7.

As an measure of worst case error, we use the standard deviation of the reconstructed point positions added to the mean position. That is, the error of an arbitrary reconstruction will be no worse than this value 68% of the time. As expected, the errors in reconstructed point location depend on θ_y, and increase as the camera is moved away from the scene. Figure 8 shows the reconstruction error in the z coordinate for distances of 0.5, 1.0, 1.5, and 5.0 meters. The error remains stable through rotations up to about 60 degrees, then begins to increase rapidly. To maintain a reconstruction error of 10%, the distance must be 1 meter or less.

4.3 Affine Invariant Error

For object recognition, the important issue is the effect of these errors on the computation of affine invariants of the structure. If we consider each point as an error ellipsoid in space, whose axes are measures of the point variance in each dimension, we can map these ellipsoids to the affine frame along with the points (see Figure 9). This ellipsoid will have its major axis in the mapped viewing direction. The error in the invariants should be distributed according to the projections of this error ellipse to the axes of the affine frame, i.e. the cosines of the angles between the mapped viewing direction and the frame's axes.

Therefore, we predict that the ratio of errors of the x, y, and z invariants will reflect the ratio of angles between these directions. When the mapped viewing direction corresponds to one axis, all of the error will be reflected in its invariant coordinate, the other two coordinates will be much more accurate. This observation suggests that a good indexing strategy is to use the affine coordinate frame

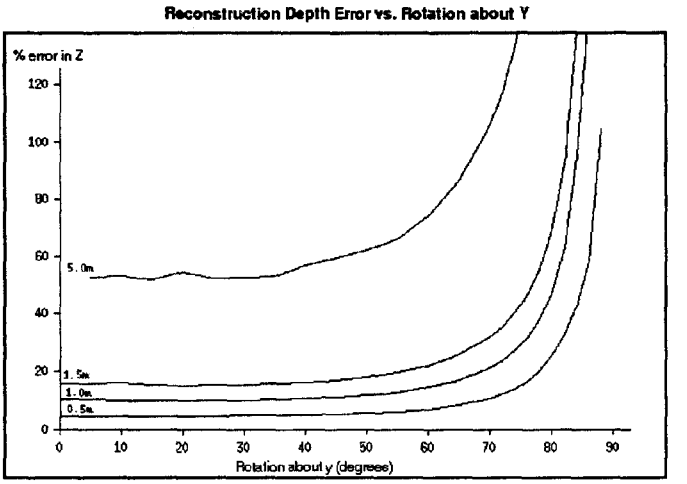


Fig. 8. Observed reconstruction error vs. y rotation at 4 camera distances.

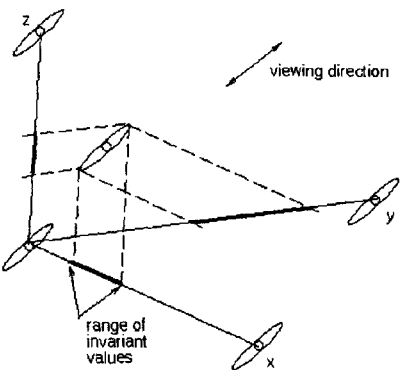


Fig. 9. Effects of error in the affine invariant frame: Each error ellipsoid from the reconstruction is mapped to the frame. The variation in the invariants depends on the projection of these ellipsoids onto the axes of the frame.

which has one axis as aligned as possible with the viewing direction in the affine frame. For five points, there are 5! affine frames but many of these combinations can be eliminated by reasoning about the convex hull of the point set. Each of the distinct frames is inserted in the model library as a separate instance of the model and has an associated invariant index.

As seen from Table 1, there is a very significant difference in indexing power between the best and worst choice of affine frame. For example, for rotation (-45, 35, 0), the first row of the table, the worst of two error deviations goes down from 10.8 for frame 1 to 3.1 for frame 4. This improvement can be realized by indexing on only two invariant values instead of three, and for the best frame.

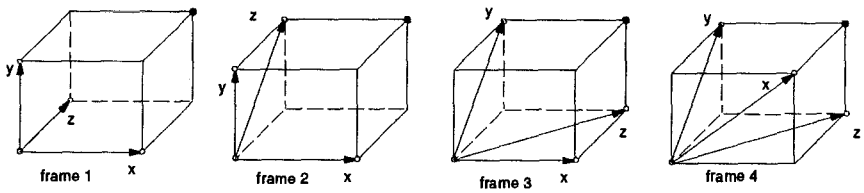


Fig. 10. Four affine coordinate frames. The points used to define the basis vectors of each frame are indicated as open circles. The location of a fifth point, which generates the affine invariants, is indicated as a black circle. The affine invariants of the fifth point for frames 1-4 are (1, 1, 1), (1, 0, 1), (1, 1, 0) and (0.5, 0.5, 0.5), respectively.

rotation angles	frame 1	frame 2	frame 3	frame 4
(-45, 35, 0)	(9.8, 10.8, 11.3)	(7.5, 3.3, 8.6)	(8.6, 8.7, 3.3)	(3.1, 3.5, 2.7)
(-63, 24, 0)	(6.3, 11.9, 8.1)	(4.4, 4.5, 5.9)	(10.1, 10.2, 4.6)	(3.5, 3.7, 0.8)
(-26, 41, 0)	(12.4, 8.1, 13.8)	(9.7, 4.8, 10.6)	(6.2, 6.4, 4.7)	(2.5, 3, 4.8)
(0, 0, 0)	(1.2, 1.6, 11.9)	(1.4, 7.7, 8.7)	(10.4, 3.7, 7.7)	(2.5, 3.5, 3.1)
(0, 22.5, 0)	(5.6, 1.0, 12.6)	(4.2, 7.9, 9.6)	(7.1, 3.8, 7.9)	(1.5, 2.9, 4.6)
(0, 45, 0)	(13.1, 2.4, 14.5)	(10.6, 8.3, 11.4)	(3.5, 3.8, 8.3)	(1, 1.9, 6.7)
(0, 67.5, 0)	(36.8, 8, 19.3)	(29.7, 9.4, 15.7)	(16.6, 3.2, 9.2)	(6.2, 2.4, 11.9)

Table 1. This table illustrates the effect of depth error on the computed affine invariants of the fifth point for each frames. The error is computed for 0.5m distance between the camera and cubes. The entries of the table show the percent deviation of a one sigma invariant coordinate error, relative to the unit distance in the affine coordinate frame. The three error deviations, one for each coordinate, are enclosed in parentheses. The Euler angles for the rotation of the viewing direction relative to the object are indicated as a triple along the left side of the table. Each column of the table corresponds to a different affine frame.

However, it should be noted that the density of models in the index space is also affected by the choice of basis. It might be the case that the improvement is not as great when model population is taken into account. Here we are assuming that models are evenly distributed over index space, regardless of choice of basis. If we consider just Frame 1, then the model density is constant. Note there is still significant error reduction for most rotations when one of the coordinates is eliminated.

5 Conclusions

A straightforward application of invariant indexing for translated structures will be effective only when the object is near enough to the camera so that perspective

effects are large. For greater viewing distances, the viewing direction is known relative to the affine frame and the effect of depth errors can be minimized. Even so, the minimum worst case deviation becomes unacceptably large as the object recedes from the camera. There are a number of strategies which can be explored to avoid the indexing errors just described.

Adjust Canonical Frames The error in reconstructed geometry is small for the coordinates "perpendicular" to the viewing direction. One can rotate the canonical frames for each object in the library so that two out of the three invariant coordinates of a point in the canonical frame are accurately known. This strategy introduces a cost proportional to the size of the model library. Still, the retrieval cost is significantly less than that for conventional model-based vision where the pose of each object in the library must be found individually. It may be feasible to define an efficient, index "rotation" hashing scheme which requires only a few computations per model.

Aligned Invariants Since the translation vector between the copies of the repeated structure is known, it may be possible to define invariants which use this fact to eliminate the effect of depth errors. For example, consider a triangle and a translated copy of the triangle in the plane. The translation epipole can act as a fourth point to define affine invariants with respect to the triangle as an affine basis. We are now studying 3D analogs of this construction to discover if similar 3D invariant results can be achieved.

References

1. Faugeras, O.D., "What can be Seen in Three Dimensions with an Uncalibrated Stereo Rig?" *Proc. ECCV*, LNCS 588, Springer-Verlag, p.563-578, 1992.
2. Hartley, R.I., Gupta, R. and Chang, T. "Stereo from Uncalibrated Cameras," *Proc. CVPR92*, p.761-764, 1992.
3. Liu J.S., Mundy J.L. and Walker E., "Recognizing Arbitrary Objects from Multiple Projections", *Proc. of Asian Conference on Computer Vision*, p.422-426, 1993.
4. Liu J.S., Mundy J.L., Forsyth D.A., Zisserman A. and Rothwell C.A., "Efficient Recognition of Rotationally Symmetric Surfaces and Straight Homogeneous Generalized Cylinders", *CVPR*, 1993
5. Mundy, J.L. and Zisserman A.,"Repeated Structures: Image Correspondence and 3D Structure Recovery", in *Applications of Invariance in Computer Vision,* Mundy, J.L., Zisserman, A. and Forsyth D.,editors, LNCS 825, Springer Verlag, 1994.
6. Fawcett R., Zisserman A. and Brady J.M., "Extracting Structure from an Affine View of a 3D Point Set with One or Two Bilateral Symmetries",*Image and Vision Computing*, 12, 9, 615-622, 1994.

MORSE: An Architecture for 3D Object Recognition Based on Invariants

J. Mundy[1], R. Curwen[1], J. Liu[2], C. Rothwell[3], A. Zissermanl[4] and D. Forsyth[5]

[1] General Electric Corporate Research and Development
Schenectady, NY 12309, USA
[2] Rensselaer Polytechnic Institute, Troy, New York 12180
[3] INRIA, Sophia Antipolis, France
[4] Oxford University, Oxford, UK
[5] D. Forsyth, Berkeley University, Berkeley, CA

1 Introduction

Over the past few years, there has been considerable interest in the application of geometric invariance to the problem of object recognition[2, 3, 4, 5, 6]. While most work has focused on the problem of discovering and characterizing new geometric invariants, several recognition systems, based on invariants have been implemented. A key example is the LEWIS[7] system. LEWIS exploits projective invariants of planar objects to enable object indexing and classification. Experience with LEWIS and its limitations, motivated the MORSE[6] The MORSE project, started in January 1994, has the goal of providing a C++ environment for the implementation of a system for recognizing 3D objects based on invariant class descriptions. MORSE embodies invariant representations for geometric classes of 3D objects such as: rotational symmetry, translational symmetry and polyhedra. The architecture is designed to support image segmentation, class-based grouping, model library management and scene reasoning. The LEWIS system has also been re-implemented using the MORSE infrastructure to provide for recognition of planar objects.

To motivate the architectural design, it will prove useful to review and contrast the key steps in object recognition for two geometric classes: rotational symmetry and structures repeated by translation. Then we will describe how these steps are mapped onto the implemented architecture. Both classes require the segmentation of image features from regions of interest. In MORSE, edgel segmentation is carried out using a modified Canny edge detector and connected edgel chains are linked topologically to form a connected network of boundary segments. The two classes have different feature grouping stages, as follows.

Rotational Symmetry Grouping When a rotationally symmetric object is projected into an image, a strong constraint is induced between the opposing boundary curves. The constraint is that the transformation which maps one side onto

[6] The acronym is Multiple Object Recognition System by Scene Entailment and is named after an Oxford detective character created by Colin Dexter. LEWIS is named after the fictional character who assists Morse in solving crimes.

the other must have a fixed line, which is the image projection of the symmetry axis. Also there is a fixed point which is the vanishing point of all lines joining corresponding points from one side to the other. Grouping proceeds by isolating concavities on each side of the object and then testing the grouping constraint for pairs of concavities. The combinatorial cost of concavity paring can be reduced using a number of quasi-invariant relations between corresponding concavities. For example, the area of corresponding concavities will be approximately equal. Also the lines joining corresponding bitangent contact points will be approximately parallel. Once a consistent axis is determined for a set of matching concavities, an invariant description of the object can be constructed by intersecting bitangent lines on the axis. The cross ratios of these distinguished points define an index for model library access.

Translationally Repeated Structures This geometric class is defined by a 3D object and a copy of the object repeated by translating the original object. The class image constraint is that all lines joining repeated points meet at a common vanishing point in the image This vanishing point corresponds to the direction of the repetition translation vector, as projected in the image. It is also the case that corresponding lines will be parallel in 3D space and therefore will be approximately parallel in the image. These constraints are used to search for a consistent set of image point pairs. The 3D geometry of the structure can be constructed using the repeated structure in analogy to a stereo reconstruction of a single object. The reconstruction is ambiguous up to a 3D affine transformation, and thus an invariant index can be constructed from 5 or more reconstructed points.

Even though the geometric constraints for the two classes are quite different, there are many common computational and database requirements. Some of the major similarities which have influenced the MORSE architecture:

Feature Organization - The grouping process requires many stages of geometric and topological representation which must be maintained for efficient access. Some structures are the shared across geometric classes, for example the representation of concavity bitangents is the same for rotationally symmetric objects and planar curved objects. It is also essential to maintain relationships among class-based groups and with respect to other constructions in the image, such as vanishing points. associate bitangent-defined concavities.

Hypothesis management - In grouping and object classification it is necessary to maintain a large number of hypotheses concerning the interpretation of feature groups and object models which are subsequently tested for validity and consistency. The architecture must support storage and maintenance of these hypotheses which involve relations between image features and between image features and models. While the details of geometric structure and grouping process control varies between geometric classes many similarities can be found. For example, it is necessary to define an ordering of the

best additional features to add to a group. Also it is necessary to eliminate failed grouping hypotheses from further consideration.

Indexing and Model Library - In all cases, the goal of grouping is to define invariant indices for accessing the model library. Models are sorted into a hash table according to invariant hash keys. For all geometric classes the model description consists of invariant feature groups and associated observable geometric features for subsequent verification.

2 The Architecture

2.1 Overview

Representation is organized into a number of layers as illustrated in figure 1. A key element of the architecture is the database, which is stratified into a number of layers. The initial layers: image, foci of attention (or regions of interest), geometric and topological features, and feature groups are all associated with a particular image. The 3D scene layer is shared across images and represents a common 3D world configuration. The model instance layer is a mixture of 3D scene interpretations and model hypotheses which only hold in a particular image.

The objects are *wrapped* in an enclosing class called a **DBWrapper** to avoid the overhead of database management on geometry, topology and feature group classes. DBWrappers for each layer are specialized to appropriate subclasses such as DBFocusWrapper, DBModelWrapper, etc. Relations among DBWrappers are implemented using the standard relational database model. That is, relations are defined as tuples of attributes, where an attribute is a pointer to a DBWrapper. For example, the vanishing point of a set of line-segments can be linked to each segment using a relational tuple containing a wrapper pointer for the vanishing point and for the line segment.

Control in MORSE is primarily distributed among the methods of individual classes. However, to maintain an overall consistency and ability to experiment with various hypothesis management strategies, a set of **KnowledgeSource** (KS for short) is defined. This approach to control is borrowed from black-board architectures[8]. It is intended that each Knowledge Source encapsulates a small, independent portion of the recognition process, so that information needed by the process can be obtained directly from the database or from a small set of parameters associated with the Knowledge Source. Each knowledge source has the ability to compute a *utility* score which indicates its suitability to execute on a given item. For example, if the best pair of line segments available are not reasonably parallel then a KS which has the task of grouping parallel line segments will report a low utility score. An overall KS scheduler selects the most appropriate KS, based on utility and tells the best KS to *execute*. The KS then executes its process using the best data available. It is also possible to set the data input for a KS and force its execution, allowing a hybrid of conventional and utility-driven control strategies.

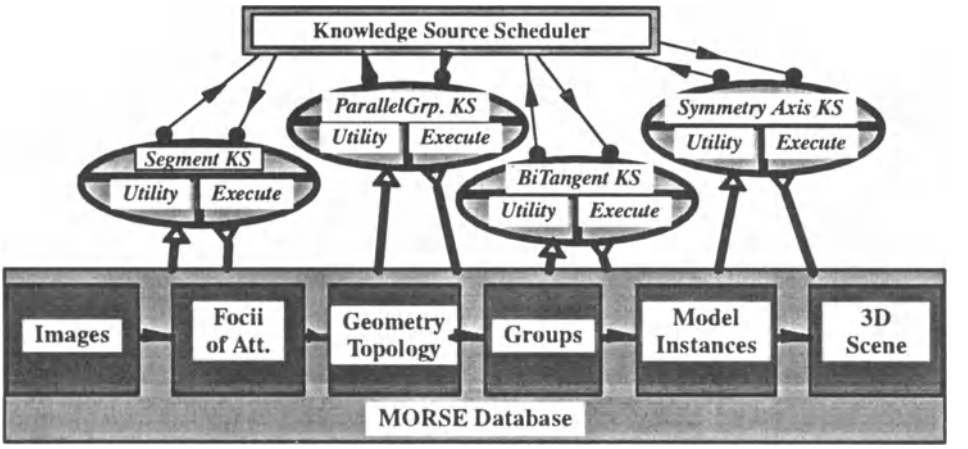


Fig. 1. The proposed architecture for object recognition. The architecture is organized around geometric *classes* which define grouping and indexing mechanisms as well as 3D scene constraints.

The model library, which consists of a set of invariant feature group classes, and associated image observables, is saved in an ascii file format developed by the Image Understanding Environment(IUE) project[1]. The ascii format, called DEX, is a generic representation for C++ class attributes and permits exchange of data between image understanding systems. In MORSE, the classes used in recognition have been augmented with DEX import/export methods.

3 Class Design

The main classes of the MORSE database are shown in Figure 2.

Class: **DBManager**

DESCRIPTION: DBManager is the root class for the domain database management system, through which all accesses to scene objects should be made. DBManager itself provides a mapping between images and object databases, so each loaded image can have its own recognized objects.

ATTRIBUTES:

hash HashTable<Image*, ImageDB*>*
The relations between Images and Image Databases.
def ImageDB* The default image database.
instance DBManager* The static instance which is returned for access to the DBManager.

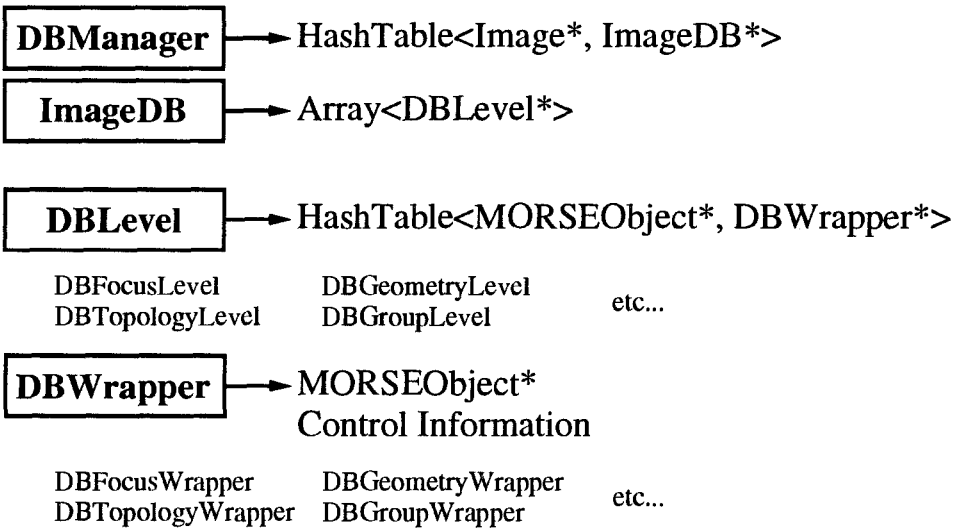


Fig. 2. The class structure of the MORSE database.

Class: **ImageDB**
DESCRIPTION: Is the repository for all of the features associated with an image. The features are organized by class types and put into *levels*.

ATTRIBUTES:

levels Array<DBLevel*>* The various levels of the database. A level is a collection of features.
image Image* A pointer to the image which is the source of the database.

Class: **DBLevel**
DESCRIPTION: A layer in the image database. Layers are nominally defined by stages of vision processing. The levels are: Focus, Point, Topology, Geometry, Group, Model, Scene. A DBLevel can be shared between ImageDB's, which is the case for the Scene level which represents the common 3D world seen in multiple images. The DBGroupLevel provides access to invariant feature group constructions as well as features associated by other relations, such as all edgel chains, *inside* some region.

ATTRIBUTES:

objects HashTable<void*, DBWrapper*>*
The storage indexing mechanism is based on a HashTable. The wrapper is

used to store information relevant to knowledge sources.
image_dbs List<ImageDB*>* The reason for the list is that images may share levels. For example, suppose that a set of feature groups is refined by stereo observations. Then two or more images have merged their databases at this level.

Class: **DBWrapper**

DESCRIPTION: Each object in the database has a database wrapper. The wrapper provides blackboard-like access to the image features without requiring any overhead on the geometric and topological structures themselves. The DB-Wrapper can be thought of as a node in a relational graph. The original design had only the relations: *::inferior* and *::superior*. It is now being extended to allow a more extensive set of relations. These new relations will allow efficient retrieval of feature groups and feasible hypotheses from the database.

ATTRIBUTES:

touch List<KnowledgeSource*>* A blackboard attribute which maintains a record that a particular KS (process) has used the given object. It is likely that this control record will eventually be handled by the relation concept, i.e., KS interactions are just another kind of relation.
protected int A counter for destructors. When **protected** =0 then the class instance can be deleted.

Class: **KnowledgeSource**

DESCRIPTION: KnowledgeSource is the base class used by the MORSE controller to encapsulate the processes by which an object representation is constructed. This class is a template for all blackboard processes, including control structures. Ideally, one KnowledgeSource does not know about any of the others.

ATTRIBUTES:

params KSParams* A standard interface for process parameters. Each Knowledge source has a set of such parameters. This design makes it possible to have global parameters which are common to all KnowledgeSources, e.g. pixel resolution.
focus DBWrapper* The dataset currently the focus of execution for the Knowledge Source. Can be defined either by maximizing a utility score over a set of possible data items or by using the method *SetFocus*.

METHODS:

double **SetFocus**(DBWrapper*) A direct forcing of the data collection which the KnowledgeSource is to employ.

double **Utility**() A value which indicates the expected benefit of executing the KnowledgeSource. For example, if a junction is well formed topologically, it can serve as a focus for polyhedral grouping.
void **Execute**() The actual running of the KnowledgeSource.

Class: **DBRelation**

DESCRIPTION: The approach follows the standard relational database model. Relations are sets of tuples of attributes, DBatr(s). The tuple is considered to be ordered which permits efficient retrieval of attributes. A DBatr is defined using an enumeration specific to a particular DBRelation. In the current design there are two types of relations, DBMultiRelation and DBSingleRelation. DBMultiRelation is designed to handle the case where an entity is related in the same way to many other entities. An example of a DBMultiRelation is the set of edgel curves belonging to the boundary of a rotationally symmetric object. DBMultiRelation is implemented by:
HashTable<DBWrapper*,List<Array<DBWrapper*>*>*>.
The DBSingleRelation is designed to handle the case where an entity is related uniquely to another entities. An example is the relationship between a line segment and its vanishing point in an image. The DBSingleRelation is implemented by: HashTable<DBWrapper*,Array<DBWrapper*>*>.

ATTRIBUTES:

Rname char* The name of the relation.
arity int The number of attributes in a tuple
key_index DBAtr The index of the key attribute
attributes Array<char*> The attribute names defined for **this** relation. The index of the array is the same for the attributes as in the case of *_tuples*.
tuples Depends on subclass Hash tables are used to provide efficient access to tuples in the relation.

4 Results

Two problem domains are presented to illustrate the capabilities of MORSE. The first domain is the grouping of outline curves from rotationally symmetric objects. The second is the grouping of parallel lines from repeated structures.

4.1 Rotationally Symmetric Grouping

Figure 3 shows results from an image of a scene containing rotationally symmetric objects. It is expensive to explore all boundary edgel chains. The first knowledge source applied, `KSGrowChains`, must form chains of interesting edgels to be passed to subsequent processes. In this example a utility function for edges

is based on the length of an edgel chain. This favors long, clean edges, such as are likely to be generated by smooth, man made objects. The knowledge source grows a long sequence of chains from the highest utility chain in the image, then "touches" the seed edge, marking it as used. In Figure 3 only 31 of the 633 chains were explored.

The next knowledge source activated, **KSFindBitangents**, finds bitangent lines from the one-chains. In this case the utility is again based on the length of the one-chain, and how close it is to being a closed region. Long closed one-chains are more likely to be outlines of volumes of revolution. The next knowledge source, **KSMatchArea**, tries to match pairs of bitangents based on the area of the curve spanned by their end points. For this purpose long bitangents spanning simple concavities were given high utility. The length factor is a condition utility, because the subsequent process of finding the central axis is more robust with a long segment of curve.

Finally, bitangent pairs are ordered with a utility which is based on how closely the areas between bitangent and curve match. The best matches are used by the expensive **KSFormAxis** knowledge source, which finds the plane projective transform between the two curve segments, and the central axis, which is the fixed line of that transform.

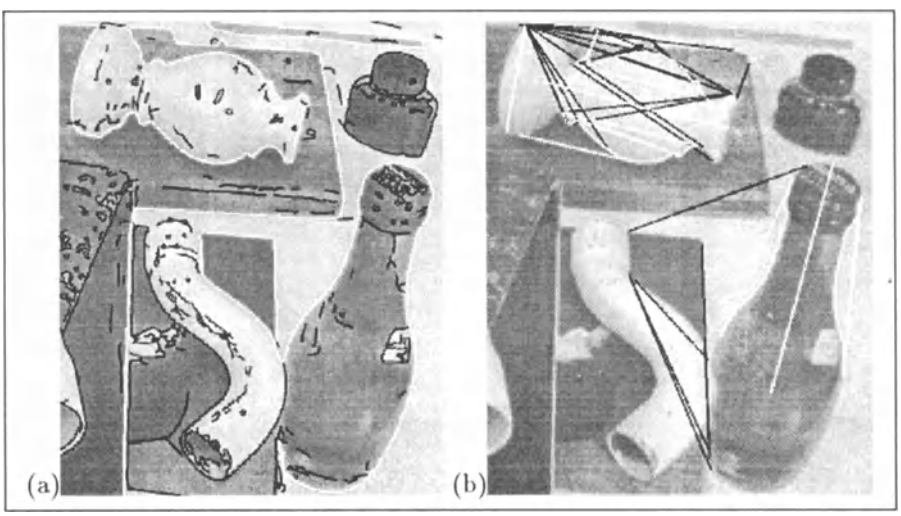


Fig. 3. Grouping volumes of revolution using utility functions to reduce search. (a) Edges with high utility shown in white. This is the subset of edges which were explored in order to complete the recognition task. (b) Bitangent lines generated for the explored edges. In white are the bitangent lines with high utility. Also shown are the two central axes recovered.

4.2 Repeated Object Grouping

Figure 4 shows the results for grouping repeated polyhedral objects. The repeated grouping problem is solved by two knowledge sources. The first, **KSSeedRepeated** matches pairs of edges to initialize groups. The second, **KSGrowRepeated** takes a seed group and adds connected edge pairs which conform to the same topology, and have roughly the same length, for both objects in the group. The group seeding knowledge source gives high utility to long, straight lines, and matches them with high utility to lines which are parallel and have similar length. The group growing knowledge source gives a high utility to groups which have good matches (orientation and length) between their corresponding edge pairs, and adds all edge pairs which are connected to the group, are similarly well matched, and maintain the topological equivalence of the two objects in the group.

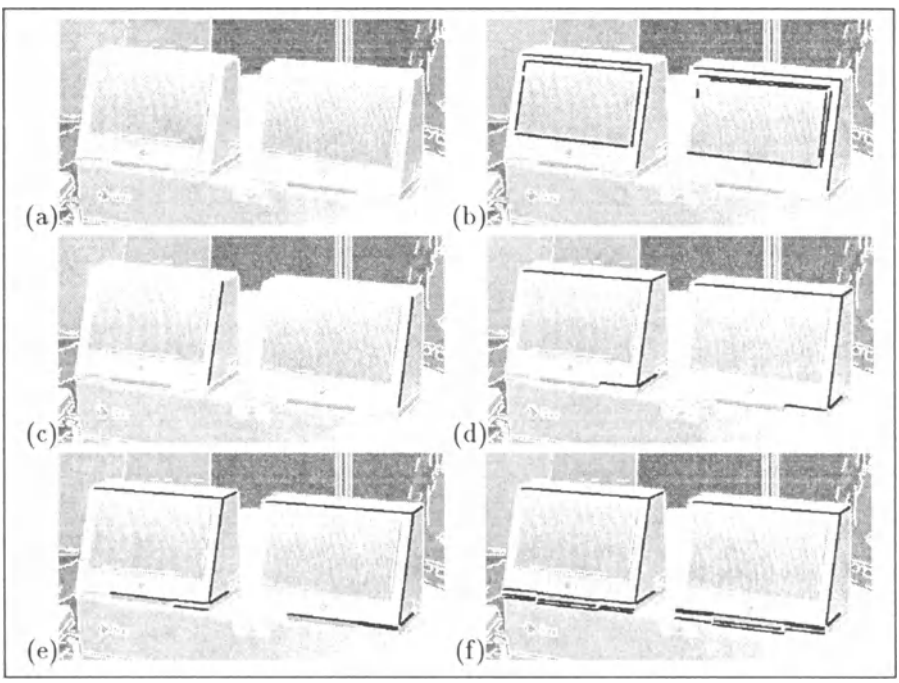


Fig. 4. Grouping repeated polyhedral structures using utility functions to reduce search. (a) The straight lines fitted to edge data. (b) Lines explored as seed edges but discarded as incorrect correspondences. (c) First correctly corresponding line pair used as a group seed. (d) Repeated groups grown from seed pair after first execution of group growing knowledge source. (e) After second execution of knowledge source. (f) After fourth and final execution of repeated group growing knowledge source.

References

1. Kohl, C. and Mundy, J.L., "The Development of the Image Understanding Environment," *Proc. CVPR*, 1994, p.443.
2. Liu J.S., Mundy J.L., Forsyth D.A., Zisserman A. and Rothwell C.A., 'Efficient Recognition of Rotationally Symmetric Surfaces and Straight Homogeneous Generalized Cylinders', *Proc. CVPR*, 1993.
3. Mundy J.L. and Zisserman A.P., *Geometric Invariance in Computer Vision.* MIT Press, 1992.
4. Mundy J.L. and Zisserman A. 'Repeated Structures: Image Correspondence Constraints and 3D Structure Recovery.' In *Applications of Invariance in Computer Vision*, Mundy, J.L., Zisserman, A. and Forsyth, D.A. (eds), LNCS 825, Springer-Verlag, 1994.
5. Pillow N., Utcke S. and Zisserman A. 'Viewpoint-Invariant Representation of Generalized Cylinders Using the Symmetry Set', *Proc. British Machine Vision Conference*, 1994.
6. Rothwell, C.A., Forsyth, D.A., Zisserman, A. and Mundy, J.L. "Extracting Projective Structure from Single Perspective Views of 3D Point Sets", *ICCV*, 573-582, 1993.
7. Rothwell, C.A., "Object Recognition Through Invariant Indexing" Oxford University Press, 1995.
8. Hayes-Roth, B., "A Blackboard Architecture for Control" *Artificial Intelligence*, vol. 26, pp. 251-321, 1985.

Object Recognition

Finding Human Faces with a Gaussian Mixture Distribution-Based Face Model

Tomaso Poggio and Kah-Kay Sung
Artificial Intelligence Laboratory
Massachusetts Institite of Technology
Cambridge, MA 02139, U.S.A.

Abstract. We present a distribution-based modeling scheme for representing and detecting human faces in cluttered scenes. A 2-Value metric is proposed for computing distance features between test patterns and the distribution-based model during classification. We present performance statistics of our overall system, and empirical results comparing the discriminative power of feature sets based on our 2-Value metric, versus similar feature sets based on other classical distribution dependent distance measures.

1 Introduction

Finding human faces automatically in a cluttered image is an important first step to a fully automatic face recognition system. It also has many potential applications ranging from surveillance and census systems to human-computer interfaces. Human face detection is difficult because there can be huge variations in the appearance of face patterns. Because many of these variations are difficult to parameterize, traditional fixed template pattern matching techniques and geometrical model-based object recognition approaches tend to perform inadequately for detecting faces. Some non-parametric approaches [4] [3] have been recently proposed for representing and detecting face patterns, but so far, they have only been successfully demonstrated on images with little background clutter.

This paper describes an example-based learning approach for finding unoccluded vertical frontal views of human faces in cluttered scenes. The approach builds a distribution-based model of frontal face views to capture the full range of permissible variations in face patterns. In order to perform pattern matching with the model, we learn a set of classification thresholds and parameters for separating "face" and "non-face" patterns, based on a set of distance measurements between the test pattern and the model.

2 System Overview and Approach

At the top-most level, our approach finds faces by searching the image for square patches of the human face (henceforth refered to as "canonical face patterns"), whose upper boundary lies just above the eyes and whose lower edge falls just below the mouth (see Figure 1(a)). The search for these face-like window patterns

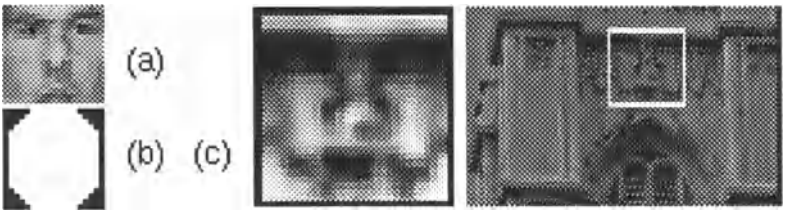


Fig. 1. **(a):** A "canonical" face pattern. **(b):** A 19 × 19 mask for eliminating near-boundary pixels of canonical face patterns. **(c):** An example of a naturally occuring pattern that resembles a human face when viewed in isolation. The same pattern is clearly not a face when viewed in the context of its environment.

is done at all image locations over multiple scales. At each image location and scale, the system classifies the local window pattern as being either "a face" or "not a face", based on its distance measurements to the face model.

Computational efficiency aside, clearly, the most critical part of our system is the algorithm for classifying window patterns as "faces" or "non-faces". The rest of this paper describes our distribution-based modeling scheme for representing and detecting canonical face patterns in images.

3 A Distribution-based Face Model

Our distribution-based modeling scheme tries to represent canonical faces as the set of all masked 19×19 pixel patterns that are canonical face views. Suppose we apply the mask of Figure 1(b) to each 19×19 image and treat each unmasked pixel as a vector dimension, then the class of all masked 19×19 images forms a vector space whose dimensionality equals the number of unmasked image pixels (283 in our case), and the set of all 19×19 pixel canonical face patterns maps to a manifold in this multi-dimensional vector space.

One can model the class of all canonical face views by recovering the manifold that corresponds to canonical face patterns, and representing the region in some tractable fashion. Figure 2 explains how we *approximate* the volume of canonical face patterns with limited data. We use a database of 4150 masked canonical "face" patterns to to obtain a coarse but fairly reliable representation of the actual canonical face manifold. We also use a carefully chosen database of 6189 non-face patterns to help refine the manifold boundaries by explicitly carving out regions around the "face" sample distribution that do not correspond to canonical face views. We shall explain how we synthesize our special database of "non-face" patterns in Section 3.2.

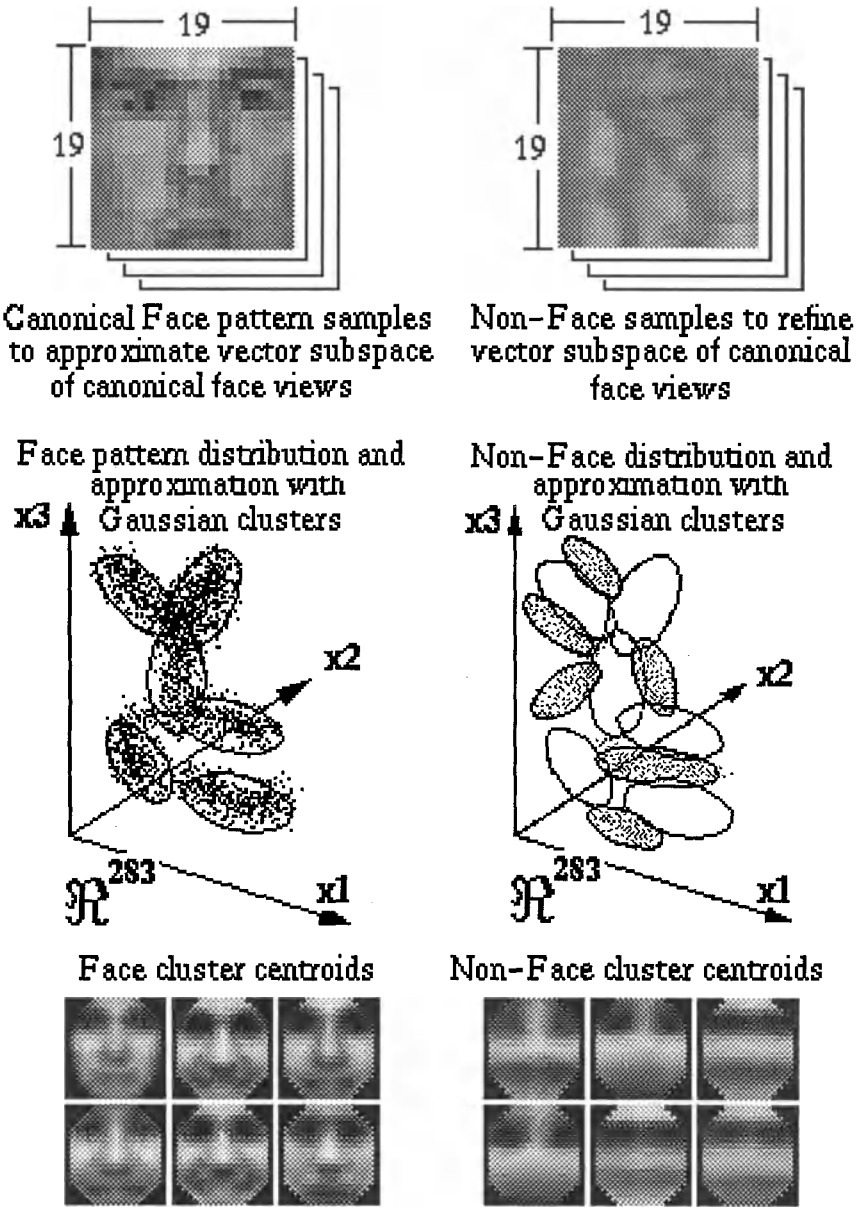


Fig. 2. Our distribution-based canonical face model. **Left Column:** We use a representative sample of canonical face patterns to approximate the volume of canonical face views in a *masked* 19×19 pixel image vector space. We model the "face" sample distribution with 6 multi-dimensional Gaussian clusters. **Right Column:** We use a selection of non-face patterns to help refine the boundaries of our Gaussian mixture approximation. We model the "non-face" sample distribution with 6 Gaussian clusters. Our final model consists of 6 "face" clusters and 6 "non-face" clusters. Each cluster is defined by a centroid and a covariance matrix.

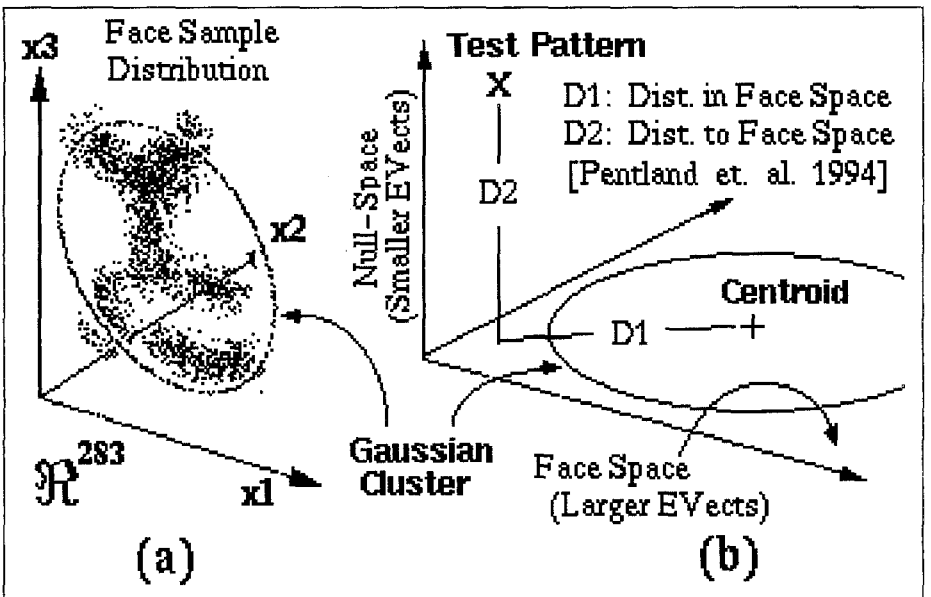


Fig. 3. (a): A single Gaussian cluster can be a very poor representation for an arbitrarily shaped "face" pattern distribution. (b): The two distance components we use in our scatter plots.

3.1 Modeling the "Face" Pattern Distribution with a Mixture of Gaussians

One can model the "face" pattern distribution by fitting the face data sample with a *single* multi-dimensional Gaussian cluster, consisting of a centroid location and a full covariance matrix. The following experiment shows that such a modeling scheme poorly describes the space of canonical face views. Using our face sample of 4150 patterns, we modeled the "face" distribution as a single multi-dimensional Gaussian cluster (see Figure 3(a)). For each face pattern in the sample, we resolved its displacement vector from the cluster centroid into two complementary components, similar to the *Distance in Face Space* and *Distance to Face Space* measures in [3] (see Figure 3(b)). We also resolved the displacement vectors of some non-face patterns from the "face" cluster centroid in the same fashion. Figure 4 shows that there is a significant amount of overlap between the face and non-face pattern distributions in this two distance component vector space, suggesting that a single Gaussian cluster representation does not separate the two pattern classes well. The distribution scatter plots are qualitatively very similar for different "Face Space" sizes.

Our approach approximates the "face" pattern distribution in a piecewise-smooth fashion using a few multi-dimensional Gaussian clusters (6 in our case). This model is reasonable as long as the actual face pattern distribution is *locally* linear, even though its *global* shape may be arbitrarily complex. The 6 clusters are

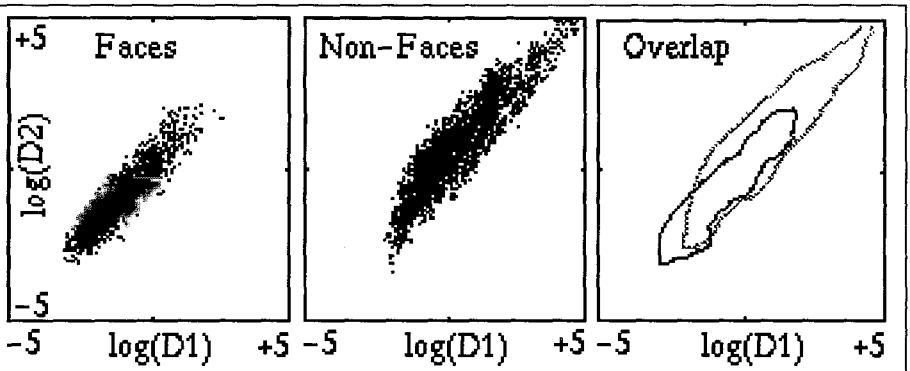


Fig. 4. Scatter plots to show that a single Gaussian cluster approximation poorly describes the space of canonical face views.

obtained by fitting 6 full covariance Gaussians to the "face" sample distribution using an *elliptical k-means* clustering algorithm (see [5] for details).

The piecewise-smooth modeling scheme serves two important functions. First, it generalizes the notion of "canonical faces" by applying a prior smoothness assumption to the observed "face" data distribution. Second, it serves as a tractable scheme for representing an arbitrary data distribution by means of a few basis functions.

3.2 Refining the "Face" Distribution by Modeling "Non-Face" Patterns

There are many naturally occuring "non-face" patterns in the real world that look like faces when viewed in isolation (see for example Figure 1(c)). Because we are coarsely representing the canonical face manifold with 6 Gaussian clusters, some of these face-like patterns may even be located nearer the "face" cluster centroids than some real "face" patterns. This may give rise to misclassification problems, because in general, one expects the opposite to be true, i.e. face patterns should lie nearer the "face" cluster centroids than non-face patterns.

To avoid confusion, we use our *elliptical k-means* algorithm to obtain 6 "non-face" clusters and their covariance matrices from a database of 6189 face-like patterns. The database was incrementally generated in a "boot-strap" fashion by first building a reduced version of our face detection system with only "face" clusters, and collecting all the *false positive* patterns it detects over a large set of natural images without faces.

Our final distribution-based model thus consists of 6 "face" clusters for coarsely approximating the canonical face pattern manifold in the image vector space, and 6 "non-face" clusters for refining the manifold by carving out non-face regions in the image vector space near the "face" clusters.

4 Matching and Classifying Patterns with the Model

To detect faces, our system resizes each candidate window pattern to 19×19 pixels and matches the resized pattern against our distribution-based face model to obtain a set of "difference" measurements. A trained classifier determines, based on the set of "difference" measurements, whether or not the test pattern is a frontal face view.

Each set of "difference" measurements is a vector of 12 distances between the test window pattern and the model's 12 cluster centroids in our multi-dimensional image vector space. One can interpret our vector of distances as the test pattern's displacement from 12 key reference locations on the canonical face pattern manifold, and hence as a crude "difference" notion between the test pattern and the entire "canonical face" pattern class.

We use a 2-Value metric to encode the distance between a test pattern and each cluster centroid. The first distance value is a directionally dependent *Mahalanobis* distance between the test pattern and the cluster centroid, in a vector sub-space spanned by the cluster's 75 largest eigenvectors. Let $\mathbf{x}$ be the column vector test pattern, $\boldsymbol{\mu}$ be the cluster centroid, E_{75} be a 75 column matrix, where column i is a unit vector in the direction of the cluster's i^{th} largest eigenvector, and W_{75} be a diagonal matrix of the corresponding 75 largest eigenvalues. The covariance matrix for the cluster's data distribution in the 75 dimensional sub-space is $\Sigma_{75} = (E_{75} W_{75} E_{75}^T)$, and the first distance value is:

$$\mathcal{D}_1(\mathbf{x}, \boldsymbol{\mu}) = \frac{1}{2}(75 \ln 2\pi + \ln |\Sigma_{75}| + (\mathbf{x} - \boldsymbol{\mu})^T \Sigma_{75}^{-1} (\mathbf{x} - \boldsymbol{\mu})). \tag{1}$$

The second distance component is a standard Euclidean distance between the test pattern $\mathbf{x}$ and its projection $\mathbf{x_p}$ in the 75 dimensional largest eigenvector sub-space. It is a robust directionally independent measure that accounts for pattern differences not captured by the first component.

$$\mathcal{D}_2(\mathbf{x}, \boldsymbol{\mu}) = ||(\mathbf{x} - \mathbf{x_p})|| = ||(I - E_{75} E_{75}^T)(\mathbf{x} - \boldsymbol{\mu})||. \tag{2}$$

We use a *multi-layer perceptron* (MLP) net to classify new window patterns as "faces" or "non-faces", based on their vector of distance measurements to the 12 cluster centroids. The net has 12 *pairs* of input terminals (for the 12 pairs of distance values) and one output node that returns a '1' for "face" patterns and a '0' otherwise. The detailed architecture of the net does not significantly affect the classifier's performance. We trained the net on our database of 4150 "face" patterns and 43166 "non-face" patterns, including the 6189 face-like patterns used for modeling the "non-face" distribution.

5 An Interpretation of our 2-Value Distance Metric

There is an interesting relationship between our 2-Value distance metric and the standard Mahalanobis distance measure that arises from fitting a full-covariance multi-dimensional Gaussian probability density to a sample data distribution. For high dimensional vector spaces, modeling a distribution with a full covariance

Gaussian is often not feasible because one usually has too few data samples to recover the covariance matrix accurately. In a d dimensional vector space, each full-covariance Gaussian requires d parameters to define its centroid and another $d(d-1)/2$ parameters to define its covariance matrix. For $d = 283$, this amounts to 40469 parameters!

One can greatly reduce the number of Gaussian model parameters by preserving only a small number of "significant" eigenvectors in the covariance matrix. Recently, Hinton et. al. [1] have used such an approach to model the distribution of hand-printed digits for digit recognition. One can show that a d dimensional Gaussian with h principal components has a covariance matrix of only $h(2d-h+1)/2$ free parameters. This can easily be a tractable number of model parameters if h is small.

In the case of modeling our face pattern distribution, the h principal components span a sub-space that preserves the most prominent pixel-wise correlations among face patterns. To exploit these pixel-wise correlations for pattern classification, one a directionally weighted Mahalanobis distance between a test pattern and the Gaussian centroid within this h dimensional sub-space, i.e. $\mathcal{D}_1$ of our 2-Value distance metric.

The orthogonal subspace spanned by the $d-h$ remaining Gaussian eigenvectors encodes pixel correlations that are less prominent, and possibly also less reliable due to the lack of training data. We regularize this subspace by assuming an isotropic Gaussian distribution of data samples. This isotropic distribution requires only 1 free parameter to describe its variance, and we use a directionally independent Euclidean measure as distance, i.e. $\mathcal{D}_2$ of our 2-Value distance metric.

We can thus view our 2-Value distance metric as two complementary components of a robust approximate Mahalanobis distance that one uses when there is insufficient data to accurately recover the full covariance matrix of a Gaussian model. The two components can be combined to form a single robust Mahalanobis distance measure $\mathcal{M}_h$ by taking their weighted sum:

$$\mathcal{M}_h = \mathcal{D}_1 + \frac{d-h}{\sum_{i=h+1}^{d} \lambda_i} \mathcal{D}_2, \tag{3}$$

where λ_i is the ith eigenvalue of the Gaussian model. In a very recent piece of work, Moghaddam and Pentland [2] discusses about an optimal estimate for high dimensional Gaussian densities, whose form is a product of two lower-dimensional marginal and independent Gaussian densities. This estimate is exactly a re-expression of Equation 3 as a probabilistic measure.

6 Results and Performance Analysis

Figures 5 and 6 show some sample face detection results by our system. "Face" window patterns are marked with an appropriately sized dotted box in the output image. Many of the faces are enclosed by mutiple dotted boxes

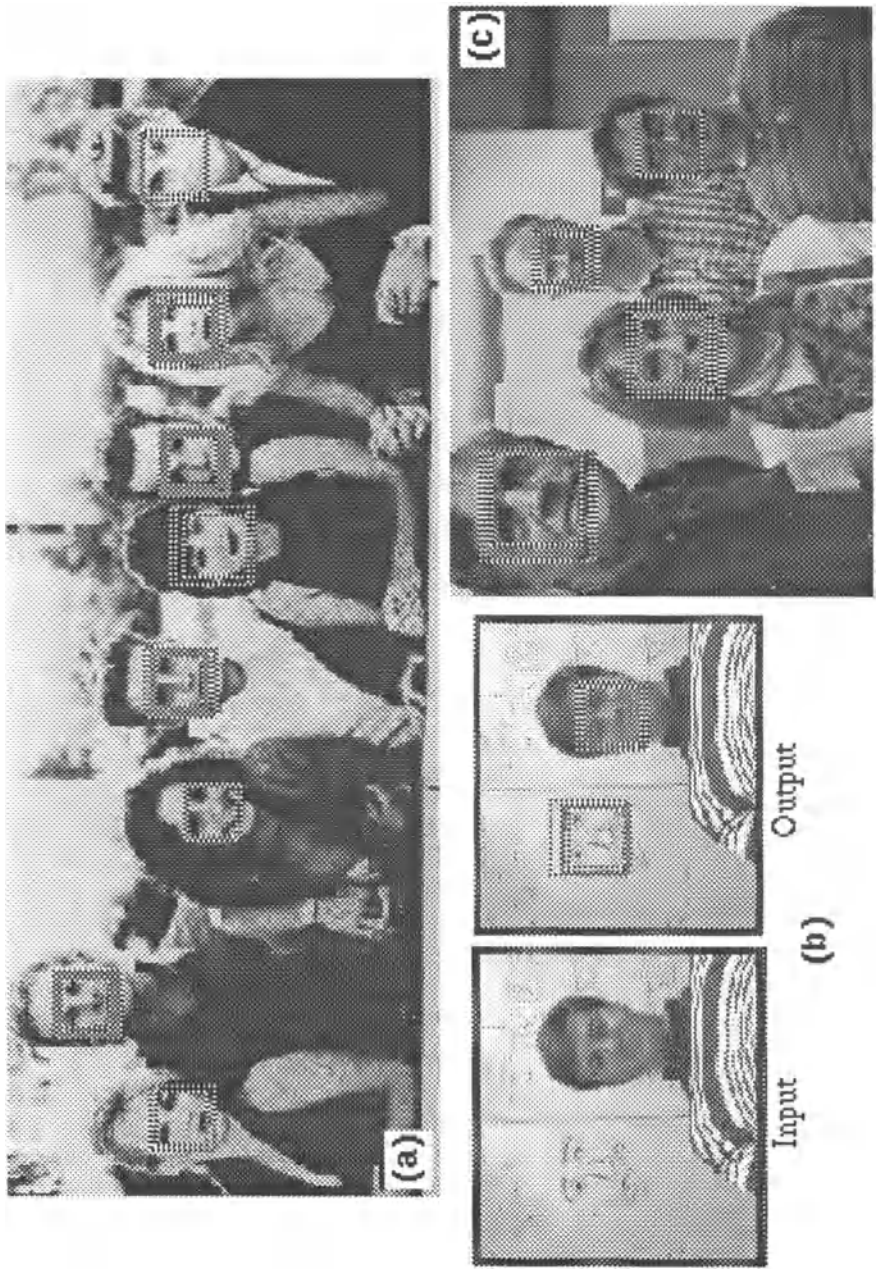


Fig. 5. Some face detection results by our system. **(a):** An image with many (8) frontal faces. The system finds all 8 faces successfully. **(b):** An input-output image pair. The same system detects real human faces and hand drawn faces. **(c):** The system finds faces successfully over multiple scales.

Fig. 6. More face detection results. The current system is trained to detect only frontal faces (Kirk and Uhura). It does not find side face profiles (Spock and McCoy) and partially occluded faces (Sulu).

because the system has detected those faces either at a few different scales or at a few slightly offset window positions. The system has a 96.3% face detection rate on a test database of 301 high quality CCD images of real people with only 3 false alarms.

We conducted the following experiment to compare the discriminative power of classification features based on our 2-Value metric, versus similar classification features based on (i) the standard Mahalanobis distance measure and (ii) its robust approximate form (equation 3). We built two new systems: The first system computes a vector of standard Mahalanobis distances between each test pattern and the 12 cluster centroids as input features to the MLP classifier, instead of the original vector of 2-Value distances. The second system computes a vector of robust Mahalanobis distances as the classifier input, with $h = 75$. We tested all three systems on a second more challenging database of 23 cluttered images with 149 face patterns, where all three systems are expected to produce more false alarms.

Table 1 compares the performance statistics of the two new systems with that of our original system. As expected, the original system with 2-Value distance features is strictly superior to the system using standard Mahalanobis distance features, in terms of having a higher detection rate and fewer false alarms. In comparing the original system with the one using robust Mahalanobis distance features, we felt that the original system produces results that are visually more appealing.

Distance Metric	Det. Rate	# False Det.
2-Value Distance	79.9%	5
Std. Mahalanobis	42.6%	5
Robust Mahalanobis	87.8%	76

Table 1. Summary of performance figures for comparing three distance metrics as classifier input features in our distribution-based modeling face detection framework.

7 Conclusion

We have successfully developed a distribution-based modeling cum example-based learning technique for representing and detecting frontal views of human faces in images. The distribution-based model captures pattern variations in face images that are difficult to parameterize using traditional pattern matching and object recognition techniques. We have also proposed a distribution dependent 2-Value metric for computing distance features between test patterns and the face model during classification. Empirically, the 2-Value metric leads to more appealing face-detection results than other classical distribution dependent distance measures, such as the standard Mahalanobis distance and its robust approximate form.

We believe that the same distribution-based modeling cum example-based learning methodology can be easily extended to take on feature detection and pattern recognition tasks in other problem domains.

References

1. G. Hinton, M. Revow, and P. Dayan. Recognizing Handwritten Digits using Mixture of Linear Models. In D. Touretzky G. Tesauro and J. Alspector, editors, *Advances in Neural Information Processings Systems 7*, San Mateo, CA, 1995. Morgan Kaufman.
2. B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Detection. In *Proceedings of the International Conference on Computer Vision*, pages 786–793, Cambridge, MA, June 1995.
3. A. Pentland, B. Moghaddam, and T. Starner. View-based and Modular Eigenspaces for Face Recognition. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 84–91, June 1994.
4. P. Sinha. Object Recognition via Image Invariants: A Case Study. In *Investigative Ophthalmology and Visual Science*, volume 35, pages 1735–1740, Sarasota, Florida, May 1994.
5. K. Sung and T. Poggio. Example-based Learning for View-based Human Face Detection. Technical Report AIM–1521, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1994.

Handwriting Recognition

Thomas M. Breuel

IDIAP, Martigny, Switzerland, and
IBM Almaden Research Center, 650 Harry Road, San Jose, CA, USA
tmb@almaden.ibm.com

1 Introduction

Recognition of handwritten text by computer can significantly reduce the cost of manual key entry in applications such as bank check processing, postal mail routing, census and poll form capture, medical documents, and many others.

In this review, I will describe the structure and performance of a handwriting recognition system. This system has demonstrated competitive performance in an evaluation by the US Census Bureau and is currently being used in the 1995 US Test Census for transcribing handwritten names, telephone numbers, and other information into ASCII text. The system operates on unsegmented handwritten input strings (i.e., there are no preprinted boxes into which respondents write characters).

Like more general machine vision tasks, handwriting recognition forces us to address questions of feature extraction, figure/ground, image segmentation, shape variation, and the integration of top-down knowledge.

2 System Structure

The system structure is diagrammed in Figure 1. There are four major processing stages.

The input to a handwriting recognition system consists of a binary (or sometimes grey-level) image obtained by scanning paper forms completed by respondents.

The *preprocessing stage* starts with a binary image of the raw input form. For each question, there is a rectangular region on the form where the handwritten response to the question can be found. The output of preprocessing is a collection of field images. Field images are slant- and skew normalized, and have any forms background removed as much as possible.

Field images are subjected to a *segmentation stage* that generates a large number of potential character subimage hypotheses. The collection of these hypotheses and their adjacency relations describe the possible segmentations of the input string into individual characters.

Each character subimage hypothesis is passed through a (character) *recognition stage* that tries to determine if the subimage represents a character (or merely represents part of an incorrect segmentation), and which are the most likely classifications corresponding to that character.

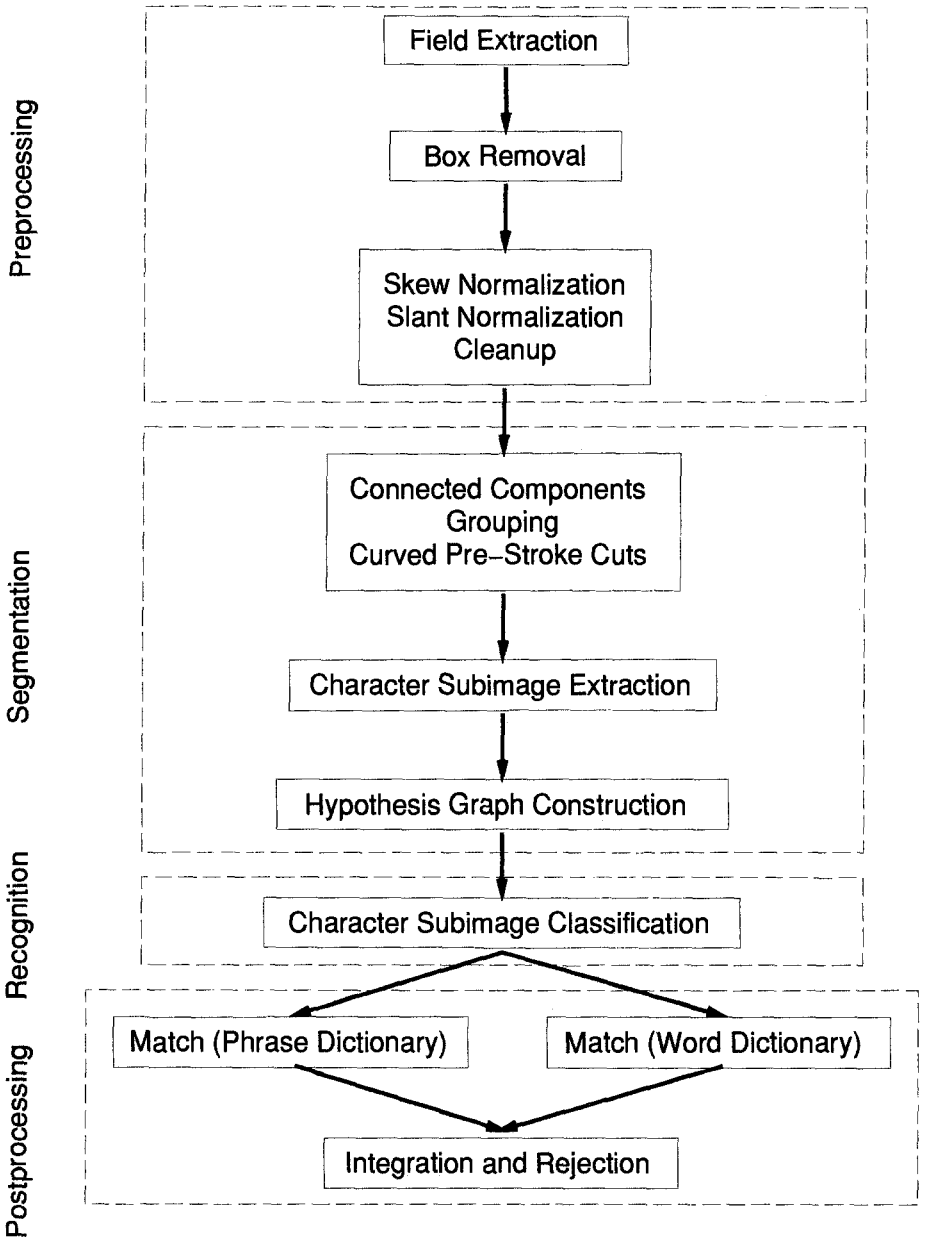


Figure 1: An overview of the system.

The *postprocessing stage* takes the classifications (and associated probabilities) from the recognition stage, together with the adjacency relations determined during segmentation. Under the constraints of a language model (dictionary or grammar), an optimal interpretation of the input is determined. In some ver-

sions of the system, multiple language models are applied simultaneously and a final integration step based on decision trees selects the most likely hypothesis obtained using different language models.

3 Statistical Foundations

Some systems for handwriting recognition use non-statistical, heuristic methods for determining the confidence in different hypotheses. This system uses probabilities to express confidences in different hypotheses; probabilities are estimated in a Bayesian framework. The approach is similar to segment based speech recognition [5].

The output of the system is a character string $W = w_1 \ldots w_n$. Under a zero-one loss function, the best decision is to output the string W with the highest posterior probability $P(W|x)$ given the input x (x represents the "signal", i.e., the image of the handwritten input). Corresponding to the string W is a segmentation S of the input image into subimages $s_1 \ldots s_n$, each of which corresponds to a character.

Of course, we do not know in advance the correct segmentation S. But, as we will see below, we can estimate the joint conditional probability $P(W, S|x)$ and remove the dependency on the segmentation S by summation:

$$P(W|x) = \sum_S P(W, S|x) \tag{1}$$

Under the assumption of context independence, the joint conditional probability distribution $P(W, S|x)$ can be approximated as: [3]

$$P(W, S|x) \approx P(W) \prod_i \frac{P(w_i, s_i|x_i)}{P(w_i)} \tag{2}$$

How good is the approximation under the independence assumption? Empirically, by examining segmented characters out of context, we find that character shapes themselves are quite independent of the shapes of their neighbors: it is very difficult from looking at the image of a character to predict much about its neighbors. But context independence is clearly violated for the contribution of the segmentation cost: if the character at position i is not segmented correctly, then it is quite likely that the character at position $i-1$ or $i+1$ (or both) is not segmented correctly either. It can be argued that the effect of this on actual error rates is small, since the main consequence is that missegmented inputs receive an extra penalty.

The above analysis describes non-constructively what constitutes an "optimal" answer W to the handwriting recognition problem given the image x. In practice, this poses a formidable optimization problem: for the set of all possible strings W (which grows exponentially in the maximal length of an answer) and among the set of all possible segmentations S of the input image into subimages, we need to compute $P(W, S|x)$ and optimize $P(W|x)$.

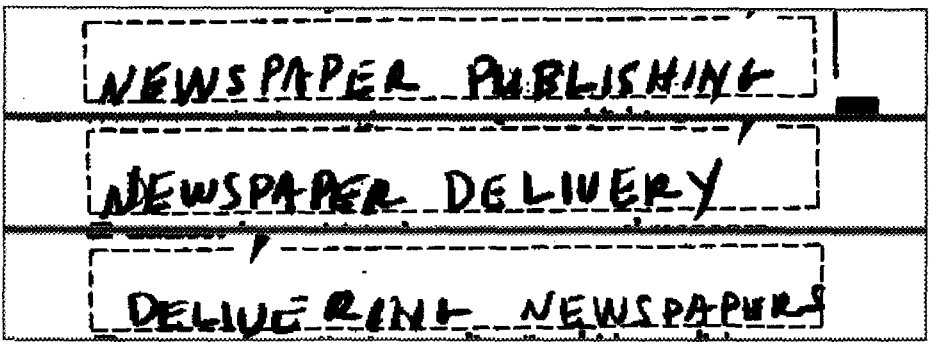


Figure 2: Examples of raw field images containing handwriting. Note that skew and forms background are still present.

Fortunately, we can make this problem computationally tractable by taking advantage of the special properties of character strings and segmentations that are plausible for handwriting recognition. In particular, we only need to consider segmentations (i.e., partitionings of the input image into subimages) that are essentially one-dimensional, are generated by considering combinations of a discrete set of "cuts" through the image, consist of non-overlapping subimages, and which cover the input image completely. Furthermore, the set of possible character strings W and their associated probabilities can be represented compactly as weighted finite state automata. Standard dynamic programming algorithms can then be used for optimizing $P(W|x)$ with relatively small computational effort.

4 Preprocessing

The purpose of preprocessing is to obtain images of the isolated handwritten responses for each field (i.e., question) from the raw forms image.

In many large volume forms processing tasks, the system designer has some influence over forms layout and scanning methods, which simplifies the preprocessing stage. In the US Census task, the layout and scanning methods were predetermined.

The input to the recognition system is a mosaic of binary images derived by thresholding from a grey level scan of census forms. Because of the comparatively poor quality of the images available in this particular application and the fact that each input image represented a mosaic, preprocessing steps were somewhat unusual.

Forms registration was based on a very robust geometric matching algorithm that identified patterns of horizontal lines present in the input images, allowing for some variation in skew. Relative to this registration, images of individual fields were extracted. Examples of such fields can be seen in Figure 2.

Fields were skew-corrected based on the rectangular dashed bounding boxes. The purpose of this skew correction step was to make the removal of the bounding

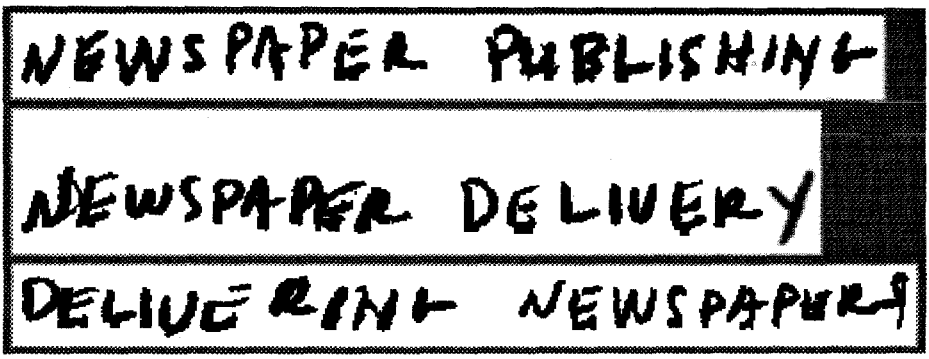


Figure 3: The output of the preprocessing step.

box and other preprinted information on the forms more robust. Preprinted information on the form, including the dashed boxes surrounding each input field were removed using morphological operations, taking care as much as possible not to remove any pixels that represent user input.

The resulting images of handwritten user input were slant corrected. Slant correction was mainly carried out to facilitate the subsequent segmentation step (the following character recognition step uses a separate, per character slant correction step). The output of the preprocessing step is shown in Figure 3.

5 Segmentation

The purpose of the segmentation step is to generate a compact representation of all the different ways in which an image of a handwritten input string can be split up into individual characters.

Field images contain no indication of where the individual characters that make up the response are located; in particular, there are no preprinted boxes for respondents to write individual characters into. The US Census Bureau is not using preprinted per-character boxes because this is thought to increase the frequency with which respondents do not fill in particular fields. Therefore, it is not possible to identify individual characters simply by location relative to the form.

Writing found on census forms consists primarily (> 79%) of handprinted characters, frequently all caps. Cleanly written strings in such a style consist of one connected component per character. In practice, however, writers produce touching characters and may even link letters together in idiosyncratic ways. In addition, characters may not only be singly connected, but can be touching in multiple places (e.g., the ink of a very closely spaced pair of upper case letters "EE", as in "EE"). Several known segmentation methods for handwriting were tried, including segmentation based on connected components, exhaustive segmentation, and valley point segmentation. However, they were found to be not well suited to the kinds of writing styles encountered on census forms.

A novel segmentation method was therefore developed. Like other methods, this method first produces hypotheses about possible boundaries between neighboring handwritten characters. Boundaries need to be curved in order to be able to cope with kerning (cases where the vertical projection of two letters overlaps partially, as in the letter pair "To"). Such curved boundaries are found at locations where strong vertical edges exist. Their shape is determined using a dynamic programming algorithm that takes into account a number of heuristic constraints. Some commonly occurring pairs of touching letters (e.g., "OO") also required special attention for segmentation.

Each pair of boundaries defines a subimage of the field image, consisting of the pixels found between the two boundaries. If there are N_b such boundaries, we potentially need to consider $\binom{N_b}{2}$ subimages. Fortunately, most of those subimages can be excluded quickly as containing a plausible character subimage, either by their aspect ratio, or by their "complexity" as measured by the number of segmentation boundaries they overlap. In practice, the number of character subimages that need to be considered is bounded approximately by $5N_b$.

The set of all segmentations S corresponds to the paths of a directed acyclic graph that has character subimages as its nodes and whose edges correspond to the relationship "character subimage x_i is to the left of, and adjacent to, character subimage x_j". We refer to this graph as the *hypothesis graph*, since it forms a compact representation of the set of all possible segmentations of the input field into characters. Furthermore, this graph has just the right algebraic properties to allow the efficient computation of the conditional probabilities in Equation 2 once per-character conditional probabilities $P(w_i, s_i|x_i)$ have been estimated.

6 Character Recognition

The purpose of the character recognition step is to estimate the per-character conditional probabilities $P(w_i, s_i|x_i)$ associated with each character subimage in the hypothesis graph. This conditional probability is a joint probability between an estimate of whether the image represents a correctly segmented isolated character ($s_i = 1$) and the class of that character (w_i).

Character recognition is done using an artificial neural network based on the backpropagation architecture [7]. It is well known that such networks applied to classification problems are good estimators of conditional probabilities (e.g., [2]). The feature set has been hand-designed and encodes local gradient information, as well as some topological properties like endpoints and interior regions.

Empirically, it was found that classifiers that perform best at isolated character recognition are not necessarily those that perform best in the handwriting recognition system. The reason appears to be that what matters most for good overall performance is robust estimates of conditional probabilities even for second or third choices; character recognizers that perform best in isolated character recognition tasks may not be optimized for giving robust estimates of probabilities of second or third choices.

7 Postprocessing

The outputs of the segmentation and character recognition steps are concise representations of the set of plausible segmentations S of the input string and the per-segment conditional probabilities $P(w_i, s_i|x_i)$. What remains to be done is the optimization of $P(W|x)$ over all possible strings W. This optimization can be carried out efficiently using a Viterbi algorithm, where we approximate $\sum_S P(W, S|x)$ by $\max_S P(W, S|x)$, an approximation commonly used for similar problems in speech recognition.

The algorithm used is a generalization of algorithms for computing the intersection of two regular languages. We can think of the hypothesis graph computed during segmentation as representing a finite, probabilistic regular language, representing all the possible interpretations of the input image in terms of different strings of characters; the probability of each such string is given by the product $\prod_i \frac{P(w_i, s_i|x_i)}{P(w_i)}$. The other regular language that plays a crucial role is the set of strings that are permissible for the given input field, represented in Equation 2 by the factor $P(W)$. Note that such algorithms rely on the essential 1D nature of the segmentation and the output representation.

8 Training

Trainable parameters for the system consist of the probabilities $P(W)$ representing the language model, and the weights of the artificial neural network used to estimate the conditional probabilities $P(w_i, s_i|x_i)$. For the US Census task, available training data consisted of 75000 raw input fields with associated transcriptions but without any segmentation information.

Because no a-priori segmentation information was available, an expectation/maximization (EM) approach under a Viterbi approximation was used for training. That is, given the input image x and its transcription W, the segmentation S is chosen that maximizes $P(W, S|x)$. This is then used to derive subimages x_i corresponding to isolated characters ($s_i = 1$, w_i representing the character class) as well as subimages representing segmentation errors ($s_i = 0$). These subimages are then used to train the backpropagation network to estimate the conditional probability $P(w_i, s_i|x_i)$. It should be emphasized that both correctly segmented characters *and* incorrectly segmented characters are used as training examples for the character recognizer, so that the joint conditional probability distribution $P(w_i, s_i|x_i)$ can be estimated both for $s_i = 0$ and $s_i = 1$.

9 Performance

To give the reader some idea of the performance of the current system, on fields like those shown in Figure 2, the system achieves field error rates at 50% rejection of approximately 6.1% when exact, letter-for-letter matches are required. The error rate at 0% rejection is 36.7%. This may appear high, but it is very

competitive with other systems applied to the same dataset [6]. It also represents commercially interesting performance, since it allows about half of the manual key entry effort to be performed by computer, resulting in substantial savings.

More interesting than the error rate itself is the sources of errors and potential areas of improvement. Perhaps the most limiting factor is an incomplete and inaccurate language model; the language model is the set of strings W that the system accepts as responses for a particular field and the associated prior probabilities $P(W)$. In fact, at 0% rejection, it is difficult to achieve error rates better than 36%, because about 36% of the handwritten inputs are not captured by the available language model for this task[1]

A good language model requires a good balance between *coverage* (roughly, the fraction of possible inputs that have non-zero prior probabilities associated with them) and *perplexity* (roughly, the number of different choices the system has to distinguish between for each input). Low coverage will keep the system from returning plausible hypotheses, while high perplexity will make it likely that implausible hypotheses are returned erroneously. In the census application, the language model $P(W)$ had to be estimated from a set of about 75000 strings, which is a very small sample.

Another significant source of rejected fields and errors is input fields that are outside the domain for which the system has been designed. About 6% of the inputs are written in a cursive style; while the system could recognize cursive input in principle, such a low fraction of cursive input means that not enough training data is available in a cursive style to adapt the system to cursive handwriting recognition. Another 2% of the input fields were not thresholded correctly at the scanner. And about 4% of the input fields contain strike-throughs or two-line entries, which require additional preprocessing steps. Altogether, at least 12% of the inputs are intrinsically unrecognizable by the current system, and would be difficult to recognize by any system for the same task.

There is also good evidence that system performance could be improved significantly by building separate models for upper- and lower-case letters. In the task described here, such training was impossible because the available training data was not labeled for upper/lower case distinctions.

10 Discussion

The system described in this paper is an interesting case study in building a handwriting recognition system and has resulted in a system with potential commercial applications. The statistical analysis should prove useful for analyzing and improving the performance of other handwriting recognition systems as well. Beyond the basic system described here, a number of task-specific improvements have been implemented that result in higher throughput and lower error rates.

[1] In order to achieve the error rate of 36.7% stated above, the system falls back on a language model with larger coverage, allowing it to perform somewhat better than expected from the coverage of the main language model.

The system has also been extended to work with numerical fields, which require somewhat different approaches to segmentation.

Perhaps more importantly, this research project was initially undertaken in an attempt to better understand some fundamental issues in visual object recognition, a path also taken by several other researchers [4]. The author believes that many of the lessons learned in the context of handwriting recognition will carry over to more general purpose vision tasks.

In order to make the handwriting recognition problem computationally tractable, we had to use a segmentation scheme that divided the handwritten into a finite set of "parts hypotheses" (character subimages). Not all of these parts hypotheses could be true simultaneously, and an optimal interpretation of the input signal required a tradeoff between spatial constraints, neighborhood relationships, and prior knowledge of plausible interpretations.

This is quite analogous to the the way visual scenes are interpreted as being composed of discrete objects and object parts (see, for example, [1]). The reasons are likely to be the same: a decomposition into discrete parts helps avoid some of the computational problems associated with more global approaches to image interpretation. The experience with handwriting recognition therefore provides additional support for the view of segmentation as an integral part of visual object recognition.

In fact, the statistical framework used for the handwriting recognition system carries over essentially unchanged to the 2D visual object recognition case.

If we can carry this analogy between handwriting recognition and visual object recognition further, we identify the following problems that need to be solved in visual object recognition:

• We need to catalog what kinds of boundaries between visual parts can occur. Occluding contours are frequent parts boundaries. But other parts boundaries occur, for example, at edge junctions in images of two joined convex 3D objects.

• We need to select and combine parts boundaries to identify image regions corresponding to complete parts. In the case of handwriting recognition, we only needed to consider pairs of boundaries and considered a "part" the image between those two boundaries. For more general vision tasks, any non-zero number of boundaries can delineate a part.

• We need to represent the spatial relationships and constraints among parts in a combinatorial structure. Important properties of the hypothesis graph and the finite state automata model used to represent $P(W)$ in the case of handwriting recognition are a certain kind of separability and locality that, in essence, allows the dynamic programming algorithm used for finding an optimal interpretation, to reuse partial interpretations of the input signal during the optimization process.

• Parts recognition can probably be carried out in a way that is quite similar to the backpropagation approach used with handwriting recognition; an indication of this is that similar methods have been applied successfully to problems like face recognition.

• In the case of handwriting recognition, the alphabet is man-made and given (although considerable natural variation occurs). For general visual object re-

cognition tasks, the "alphabet" may be partially innate and partially acquired through unsupervised learning.

Of course, there has been considerable work in computer vision addressing issues of segmentation, Bayesian classification, and visual learning. However, it is hoped that the analogy to handwriting recognition will provide a useful framework for thinking about and organizing such work.

References

[1] I. Biederman. Human image understanding: Recent research and a theory. *Comp. Vis. Graph. Im. Proc.*, 32:29–73, 1985.

[2] H. Bourlard and C. J. Wellekens. Links between Markov models and multilayer perceptrons. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1, pages 502–510, San Mateo, CA, 1989. Morgan Kaufmann.

[3] Thomas M. Breuel. A system for the off-line recognition of handwritten text. Technical Report 94-02, IDIAP, Martigny, Switzerland, 1994. submitted to ICPR'94.

[4] S. Edelman, S. Ullman, and T. Flash. Reading cursive handwriting by alignment of letter prototypes. *International Journal of Computer Vision*, 5:303–331, 1990.

[5] Hon C. Leung, I. Lee Hetherington, and Victor W. Zue. Speech Recognition using Stochastic Explicit-Segment Modeling. In *EUROSPEECH 91. 2nd European Conference on Speech Communication and Technology Proceedings*, Genova, Italy, 1991. Instituto Int. Comunicazioni.

[6] *The Second Census Optical Character Recognition System Conference*, February 1994.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(9):533–536, October 1986.

An Integrated Model for Evaluating the Amount of Data Required for Reliable Recognition

Michael Lindenbaum *

Computer Science Department, Technion,
Haifa 32000, ISRAEL

Abstract. Many recognition procedures rely on the consistency of a subset of data features with an hypothesis, as the sufficient evidence to the presence of the corresponding object. The performance of such procedures are analyzed using a probabilistic model and provide expressions for the sufficient size of such data subsets, that, if consistent, guarantee the validity of the hypotheses with arbitrarily prespecified confidence. The analysis focuses on 2D objects and on the affine transformation class, and is based, for the first time, on an integrated model, which takes into account the shape of the objects involved, the accuracy of the data collected, the clutter present in the scene, the class of the transformations involved, the accuracy of the localization, and the confidence required in the hypotheses. Most of these factors can be quantified cumulatively by one parameter, denoted "effective similarity", which largely determines the sufficient subset size.

1 Introduction

Model-Based object recognition and localization are fundamental tasks of Computer Vision. In localization, one observes a scene, looks for evidence to the presence of a known object in it and tries to hypothesize its position. In recognition the object is included in a known library but is otherwise unknown, and its identity as well as its position are to be determined.

Objects look different in different images, due to their location relative to the camera (referred to as "pose" or "transformation"). Most recognition approaches rely, at least implicitly, on an "hypothesize and test" approach: they hypothesize a particular object and a particular transformation, calculate the expected image, and check its similarity to the actual image. Recognition methods differ mainly in the way they draw the hypotheses. The verification process is usually similar and is based on using the fraction of the image features (usually edge points), consistent with the boundaries of the hypothesized object, as a measure to its plausibility. (Some exceptions are described in [SG93, B93].)

Many reasons may cause the verification process to fail: the data features in a consistent subset may correspond to another, incorrect, object or may just be created by the clutter in the image. Even if they correspond to the true object,

* email address:mic@cs.technion.ac.il

they may not restrict the pose of the object sufficiently, and the hypothesized object instance may still be associated with some intolerable localization error. All these reasons are addresses by our analysis, which considers the different kinds of recognition failures with respect to the combined effect of the object's shape, the class of transformation allowed, the measurement accuracy, the clutter in the image, and the presence of similar objects in the background. We focus on two dimensional rigid objects and instances of them obtained using the Affine, Similarity, and Euclidean transformations, and provide expressions for the size of consistent data subsets, which guarantee the validity of instance hypotheses within arbitrary pre-specified confidence.

The amount of information required for reliable verification was considered in a few papers: Grimson and Huttenlocher analyzed the possibility that a subset of "noise data features" will give a false evidence for the presence of an object in the scene [GH91a]. Similar methods are used in the analysis of the reliability of Geometric Hashing technique under various noise models [GHJ92], [SG93]. Maybank used the cross ratio invariant function distribution to predict its reliability as an evidence for object presence [M93]. Lindenbaum analyzed the effect of the object's shape on the recognition difficulty and quantified it by two parameters, Self-similarity and similarity, that characterize the difficulty of localization and recognition, respectively [L95]. More recent papers of Ben-david and Lindenbaum [BL93, LB94a, LB94b] consider this problem from a combinatorial viewpoint, and derive, using learning theory tools, upper bounds on the probability of drawing incorrect hypotheses.

Our work is different from previous contributions in several aspects: It is the first approach that consider complicated scenes in which both the effects of object similarities and clutter act to make recognition less reliable. A second crucial difference is that unlike some other papers, here we do not examine neither the probability that a particular hypothesis fails nor the probability that some particular hypothesizing mechanism fails, but rather the probability that any hypothesis consistent with k data features fails. The later probability is naturally higher and sets an absolute bound on the error irrespective of the particular algorithm and in particular of the way it chooses its hypotheses. In this sense, our approach is algorithm independent. We make no assumptions on the objects, except an upper bound on their perimeter, and unlike other methods, the objects are neither required to be polygonal [GH91a], nor assumed to be represented only by a collection of feature point [GHJ92], [SG93]. The derivation is rigorous and no assumptions, except the modeling of the data itself, are made.

The analysis is based on representing the class of instances corresponding to a model object and a group of transformations, as members of a metric space, and quantifying the variation of the instances by a metric cover. This short version only presents the theoretical predictions, explains them and compares them to empirical results. The analysis is omitted and may be found in [L93].

The rest of the paper presents the probabilistic model, on which the analysis relies, the bounds on the required consistent data set size, and some simulation results, which supports the analytical derivation.

2 A framework for recognition tasks

We consider recognition processes that are composed of a data collection stage followed by an interpretation stage. Most recognition processes, with the exception of those which use active sensing strategies, are consistent with this description. These stages are described in the next two sections.

2.1 The data collection model

The first stage uses an edge detection process to extract a set S of feature points which hopefully lie close to the objects boundary. This process is modeled by assuming that the data features originate either by the boundaries of objects that are present in the scene, or by the general clutter (noise) in it, and are drawn according to a piecewise uniform distribution described as follows: Let ∂V_t be the boundary of an instance of the object V, corresponding to a transformation t. We assume that data points extracted from this boundary are independently sampled according to a uniform distribution, inside

$$V_t^{\Delta} = \{ r \mid \exists s \in \partial V_t \ s.t. \ ||s - r|| < \Delta \}, \tag{1}$$

to which we refer as *extended boundary*, and is just the morphological dilation of ∂V_t with a circular structuring element. (Changing the boundary originated distribution and the corresponding analysis and results, to one corresponding to Gaussian error model, is straightforward.)

Thus, in regions that belong to some extended boundary of some object in the image, the distribution density, f_b, is constant and higher than the clutter density, f_c, in the rest of the image. $f_c < f_b = s \cdot f_c$. ("s" stands for the saliency of the boundaries over the image clutter.) We further assume that the objects observed in the image are in relative random locations, and denote the fraction of area occupied by their extended boundaries by α. For an $l \times l$ image, the constraint $l^2[\alpha \cdot f_b + (1-\alpha) \cdot f_c] = 1$, obtained by integrating the density over the image, readily follows . We refer to this image model, which partially specifies the statistical scene properties, as *mixed data model* and denote it $MDM(N, s, \alpha)$, where $N = ||S||$.

The MDM model addresses important issues such as the dependence of the data on the objects in the scene, the uncertainty on the observed part of the object and the inaccuracy of the measurements. The images produced by the model have much in common with real edge images, although they do not look exactly like them (see Figure 1). To our best knowledge, this is the most general model considered so far for general edge images, which lends itself to rigorous analysis without approximations.

2.2 The interpretation stage

In the second stage the data collected is combined with prior knowledge, and is interpreted, to yield an hypothesis on the identity and pose of the objects

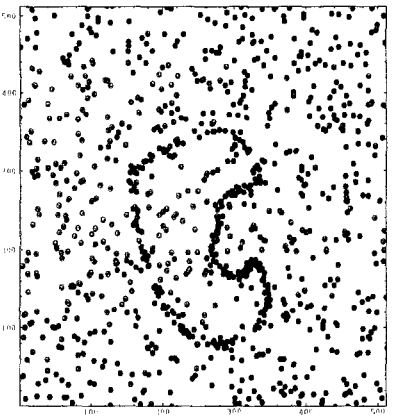


Fig. 1. A synthetic collection of data features created according to the *mixed data model* (see definition later), assuming one object is present in the scene (parameters $s = 5, \alpha = 0.085$). Note the accumulation of points near the boundaries of the object in the image center.

in the scene. We refrain from referring to any particular method for inferring the hypothesis about the object identity and pose, but only assume that every hypotheses V_t, consistent with k data features (i.e. satisfying $||S \cap V_t^{\Delta}|| \geq k$), may be drawn.

Most practical recognition algorithms include another crucial stage, of drawing the candidate hypotheses out of some discrete version of the hypothesis space. They are usually obtained either by explicit discretization of the transformation parameters (e.g. Hough Transform) or some data-driven implicit discretization (e.g. alignment or Geometric Hashing). Here, we do not consider any such discretization but assume that every hypothesis that satisfies the above criterion is accepted. Therefore, in our analysis of false hypotheses we consider all hypotheses made by any particular algorithm, and many more, which makes our results more conservative than those derived in the context of particular algorithms.

3 The consistent subset size required for reliable Localization

This section presents an expression for the sufficient size of a data subset, that, if consistent with an affine instance of the object V, guarantees that this instance is a sufficiently good hypothesis of the correct object instance. Similar results exist for the less general Similarity, Euclidean and translation transformation classes. By the title "sufficiently good hypothesis", we mean that it is an instance of the correct object, and that its pose if close enough (in a quantitative sense) to the pose of the correct instance. The result is derived for the localization procedure, which interpret the image data by hypothesizing instances of one particular object. The next section generalizes the results to recognition, in which instances of several objects, included in a known library, are considered.

Theorem 1. *Let S be a set of N random features drawn in an $l \times l$ image according to the* mixed data model, *$MDM(N, s, \alpha)$. Consider a localization procedure which tries to localize the object V under the affine transformation group $\mathcal{A}$ with required localization precision d_0, but outputs only hypotheses, V_t for which $||S \cap v_t^{\Delta}|| \geq k$. Then, the localization procedure is guaranteed, with confidence $1-\delta$, not to produce any false hypothesis, if*

$$k \geq \frac{1}{log_e(v+1)} \left[vNp(s, \alpha, r_{total}) + 6 \log_e \frac{evNs}{6(1+\alpha(s-1))} + \log_e c(\mathcal{A}, V, \Delta) + \log_e \frac{1}{\delta} \right], \tag{2}$$

where e is the natural logarithm basis, v is any positive number, $c(\mathcal{A}, V, \Delta)$ is a constant which depends on the transformation class, and $p(s, \alpha, r_{total})$ is an *effective similarity* parameter, which characterizes the similarities of all objects related to the task. All parameters are described below.

Proof. The proof, which relies on representing all instances as members of a metric space, may be found in [L93].

Example 1. Before explaining the different parameters that determine the localization task difficulty and the expression described above, the meaning of the sought for sufficient subset size is illustrated relative to a particular algorithm: a continuous Hough transform. Let $L(\rho, \theta)$ denotes the line $\{(x, y) | x cos\theta + y sin\theta = \rho\}$. Consider an image containing no lines but only N uniformly distributed random data features. Every image point (x_i, y_i) corresponds to a parameter region $K_{(x_i, y_i)} = \{(\rho, \theta) | \exists (x, y) \ s.t. \ (x, y) \in L(\rho, \theta) \ AND \ ||(x, y) - (x_i, y_i)|| \leq \Delta\}$, which contains all parameter pairs corresponding to lines passing Δ–close to that point. The Hough transform essentially looks for the maximum of $H(\rho, \theta) = \sum_{i=1}^{N} K_{(x_i, y_i)}$. This may be be interpreted as looking for an instance of a line object assuming a transformation involves rotation and 1D translation. The scene contains no line, implying that this maximum should not be too high. The sufficient subset size provided here bounds the maximal value of $H(\rho, \theta)$, which does not necessary correspond to any (ρ, θ) value included in a grid, but clearly bounds the maximal value obtained on any grid (the case of a particular algorithm: the regular, discrete, Hough transform), as well as the value of $H(\rho, \theta)$ for any particular (ρ, θ) pair (the case of a particular hypothesis).

The expression (2) is admittedly complicated and depends on many parameters. We shall go over them now, one at a time, explain their meaning and comment on their intuitive and quantitative implications.

The confidence $1-\delta$ is the probability of not getting any accidental consistency. Higher confidence (lower δ) requires more data (higher k).

The transformation class effect - A more general transformation class (with more degrees of freedom), implies, intuitively, that there are "more instances" and thus a higher probability for false consistency. The quantitative effect of this higher instance number depends both on the transformation class and on the image properties and is given by the second and third terms of (2). The constant $c(\mathcal{A}, V, \Delta)$ depends on the object's perimeter L and is

$c(\mathcal{A}, V, \Delta) = \frac{1}{2} \cdot 6^6 (l + L^\Delta/2)^2 (L^\Delta)^6 L^4 l^{-12}$ ($L^\Delta = L + 2\pi\Delta$). Lower constants were derived in [L93] for less general classes.

The effect of object similarities - Intuitively, if the scene contains some instance $W_{t'}$ of an object W which is very similar to some instance V_t of the hypothesized object V, then it is likely that data features from the boundary of $W_{t'}$, will cause a false evidence of substantial size for V_t. This intuitive observation is transformed into quantitatively meaningful similarity parameter, which is defined as follows: Let $W_{t'}$ be an instance of the object W which lies in the scene. Then, for every one of the instances V_t, the probability that a data feature from $W_{t'}$ is consistent with V_t is

$$Prob\{x \in V_t^\Delta \,|x \in W_{t'}^\Delta\} = \frac{area(V_t^\Delta \cap W_{t'}^\Delta)}{area(W_{t'}^\Delta)} \leq \max_{t,t' \in T} \frac{area(V_t^\Delta \cap W_{t'}^\Delta)}{area(W_{t'}^\Delta)}. \tag{3}$$

The area ratio $\max_{t,t' \in T} \frac{area(V_t^\Delta \cap W_{t'}^\Delta)}{area(W_{t'}^\Delta)}$ is denoted $r_{max}^{W,V}(\Delta)$, and is named *Similarity* [L95]. It quantifies the similarity of the objects regarding their shape, the allowed transformations, and the observation precision Δ. A low, close to zero, value of the similarity parameter indicates that the instances of V and W are always very different visually and that a small number of data features probably suffice to discriminate between them. A high, close to one, value indicates, on the other hand, that there are instances of V and W which are very similar, which makes the discrimination between them difficult. Note that if the T is identical to the affine transformation class, then the similarity may be as high as one, corresponding to the intuitive observation that sometimes (e.g. when the transformation involves extreme skew) it is impossible to discriminate between any two objects. For $V, W_1, W_2, \ldots$ being the objects that may appear in the scene, then the effect of their similarities is quantified by specifying the *total similarity* parameter as

$$r_{total} = \max[r_{max}^V(\Delta, d_0), \max_i r_{max}^{W_i}(\Delta)].$$

(The first term, $r_{max}^V(\Delta, d_0)$, denoted *self-similarity*, generalizes the notion of object symmetry and quantifies the difficulty of localizing the object V with precision d_0. Its definition requires the quantification of localization quality and may be found in [L95, L93].)

Note also that the *mixed data model* parameters strongly affect the object similarities effect. If, for example, the saliency is 1, implying that the boundary does not produce data features more than the background, then similarity has no effect. When the *mixed data model* parameters are taken into account, the probability that a data feature is consistent with some particular instance of V is given by

$$p(s, \alpha, r_{total}) = \frac{1}{l^2} \max_{t \in \mathcal{A}} area(V_t^\Delta) \left[1 + \frac{r_{total}\,(s-1)\,(1-\alpha)}{1 + \alpha(s-1)}\right]. \tag{4}$$

defined as the *effective similarity* parameter. If this probability is large, so is the probability for a false consistency. Therefore, is the *effective similarity* is high, the sufficient set size should increase.

The constant v - This parameter is induced by our analysis method and can be arbitrarily set to be any positive constant. This means that the expression (2) provides a set of sufficient set sizes which are all equally valid. Instead of arbitrarily setting it to some value, it is better to calculate (2) for a range of v values and to choose the best (lowest) one.

The effective similarity and the associated consistent subset size increase with the presence of objects in the scene and with increasing saliency. Note that the claim assumes that the objects expected in the scene are known. Actually, we just have to know the similarity between V and the most similar scene object. ((Note that the objects expected in the scene are usually different than the set of library objects, on which the hypotheses are based in the general recognition task.) This is natural to demand, because very similar objects to V lead inevitably to false hypotheses. If this information is not known in advance, then our claim should be interpreted as a conditional one. That is, as a relation between the total similarity that characterizes the localization task and the sufficient data set size.

4 From Localization to Recognition

A simple extension of our result would be to consider the general model-based recognition task. Let $\mathcal{L} = \{V_1, V_2, \ldots, V_{\|\mathcal{L}\|}\}$ be a library that contains several objects. Intuitively, more objects imply "more object instances" and a higher probability to err. Therefore, the consistent data subset size should increase. We proved that the required subset size is the same as give by (2) except that the $\log_e c(\mathcal{A}, V, \Delta)$ term changes to $\log_e \sum_i c(\mathcal{A}, V_i, \Delta)$ and the maximal effective similarity, over all object in $\mathcal{L}$, replaces the effective similarity (4).

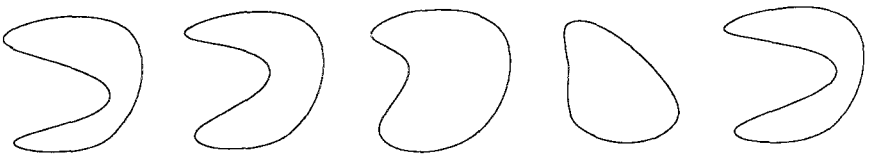
This is not surprising and was shown already in the context of clutter-less scenes for different object models [LB94a, L95]. Moreover, it is known by experiments that some recognition algorithms run in time proportional to the logarithm of the library size [GTM89]. We have shown here that the amount of information required to succeed in the task increases with logarithmic rate as well.

5 The necessary consistent subset size

As claimed above, if an instance is consistent with a subset of data features, the size of which is given by (2), then, it is guaranteed, that this object is indeed in the imaged scene with high confidence. Much simpler arguments imply that a consistency of at least

$$k_{necessary} = \min_{\sum_{i=k}^{N} \binom{N}{k} p_0^k (1-p_0)^k < \delta} k \tag{5}$$

($p_0 = p(s, \alpha, r_{total})$) data features is necessary for such conclusion.



a b c d e

Fig. 2. The objects used for the simulations: The scene was based on either object a, b, c, or d, and the hypotheses were based on object e. For the Translation class, the similarities between the scene objects and the model were $r_{max}^{a,e} = 0.88, r_{max}^{b,e} = 0.71, r_{max}^{c,e} = 0.64$ and $r_{max}^{d,e} = 0.40$ respectively.

6 Simulation results

The bounds are compared to the actual performance of one of the simplest recognition methods, the Hough transform. This algorithm, which we chose because of its simplicity, is too slow and space demanding for testing the predictions related to the more complicated transformation classes, and our simulations are thus restricted to the simpler translation and Euclidean classes. Yet, these results are enough to show, at least for these transformations, the validity of the expressions derived and also the gap that still exist between the predictions and the actual values.

Random synthetic scenes, similar to the example given in Figure 1, were constructed according to the mixed data model, each image containing one of the objects a, b, c or d (see Figure 2). The false hypothesis that it is some instance of the object e that is in the scene was examined (discrimination task). For every scene model and saliency, 5000 random images were created and the statistics of the maximal vote in the Hough accumulator was evaluated.

A representative subset of the results is shown in Fig. 3 where the consistent set size was plotted as a function of the effective similarity. The three points corresponding to a particular effective similarity are the predicted sufficient set size (always the upper point), the predicted necessary set size (always lower), and the actual (measured) sufficient set size (always middle). (In this plot, there are three point triplets for every one of the four scene objects, and the corresponding saliencies are 1, 10 and 50.) Note that all bounds are valid for all the experimental situations tested. There is always a gap however between the upper bounds and the corresponding actual results meaning that these bounds are not tight. On the other hand, the lower bounds are tighter and usually almost coincide with the actual measurements. It is apparent that it is not the similarity or the saliency alone which determine the sufficient subset size but rather their cumulative effect as represented by the effective similarity parameter.

More detailed account of these experiments, including all numerical results, as well as a description of other experiments (scenes which contain additional straight distractors, experiments with the Euclidean transformations class, etc.) are reported in [L93].

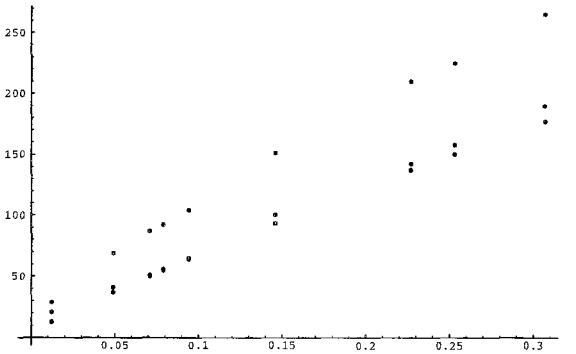


Fig. 3. The predicted bounds on the sufficient set size (upper and lower points) and the actual sufficient set size as determined by experiments (middle points), plotted against the effective similarity which characterizes the scene.

7 Discussion

In the course of the research on object recognition, many intuitive and observations were made, and were supported by the vast experimentation in the field. A partial list of them would be: "similar objects are harder to discriminate", "finer localization fails more often" and "larger candidate object library requires more data but not too much". This paper provides quantitative support for the above intuitive observations, in the form of expressions for the amount of data extracted from the image, that, if consistent with some object hypotheses, provides a sufficiently reliable evidence for the existence of this object in the imaged scene. These expressions explicitly show how the amount of information required for success is affected by the following various parameters:

- The amount of clutter in the image,
- the transformation class,
- the similarity between objects and their symmetry (self-similarity),
- the accuracy of the data,
- the localization precision required, and
- the confidence we would like to have in our hypotheses.

To use these results for recognition procedures, one should characterize the recognition setting by the MDM parameters and by the similarities and self-similarities of the related objects. Then, the expressions (4,2) and (5) can be used to direct the decisions of the verification stage. Another application is to evaluate the difficulty of recognition tasks, and thus to infer the performance of a recognition procedure that were tested in a particular setting, when conditions are changed.

An natural extension, which is considered now, is to find the sufficient subset size data features that carry additional information such as the curve slope or

curvature. Although the sufficient size derived here obviously holds also for the richer data, we hope to get smaller and tighter bounds.

This paper may be considered as analyzing the "information complexity" of recognition tasks. Another interesting extension is to find the "time complexity". That is, to find the sufficient and necessary bounds on the time required for finding the correct hypothesis, if the information is sufficient for this task. This was done for specific algorithms (e.g. see Cass [C92]) but more inherent limitations were not explored yet.

References

[BL93] Ben-David, S. and M. Lindenbaum, 1993, "Localization vs. Identification of Semi-Algebraic Sets", Proceedings of the 6th ACM Conference on Computational Learning Theory, pp. 327-336.

[B93] Breuel, T.M., 1993, "Higher-order Statistics in Object Recognition", Proc. CVPR, pp. 707-708.

[C92] Cass, T.A., 1992, "Polynomial Time Object Recognition in the Presence of Clutter Occlusion, and Uncertainty", Proceedings of ECCV, pp. 834-842.

[GTM89] Gottschalk P.G. , J.L. Turney, and T.N. Mudge, 1989, "Efficient Recognition of Partially Visible Objects Using a Logarithmic Complexity Matching Technique", *Int. J. of Rob. Res.*, **8**(6), pp. 110-131.

[GH91a] Grimson, W.E.L., and D.P. Huttenlocher, 1991, "On the Verification of Hypothesized Matches in Model-Based Recognition", *IEEE Trans. on Pattern Analysis and Mach. Intel.*, **PAMI-13**(12), pp. 1201-1213.

[GHJ92] Grimson, W.E.L., D.P. Huttenlocher, and D.W. Jacobs, 1992, "A study of affine Matching with Bounded sensor error", *Second Europ. Conf. Comp. Vision*, pp. 291-306.

[L95] M. Lindenbaum, 1995, "Bounds on Shape Recognition Performance", *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-17, No. 7, pp. 666-680.

[L93] M. Lindenbaum, 1993, "On the Amount of Information Required for Object Recognition", Proceedings of the 12th International Conference on Pattern Recognition, Vol. I, pp. 726-729, .1994. Also CIS report 9329, Technion, November 1993.

[LB94a] M. Lindenbaum and S. Ben-David, 1994, "Applying VC-dimension Analysis to Object Recognition", Proceedings of the 3rd European conference on Computer Vision, pp. 239-240.

[LB94b] M. Lindenbaum and S. Ben-David, 1994, "Applying VC-dimension Analysis to 3D Object Recognition from Perspective Projections", Proceedings of the 12th National Conf. on Artificial Intelligence (AAAI), pp. 985-990.

[M93] Maybank, S.J. ,1993, Probabilistic Analysis of the Application of the Cross Ratio to Model Based Vision", *International Journal of Computer Vision* 16, pp. 5-33.

[SG93] Sarachik, K.B., and W.E.L. Grimson, 1993, "Gaussian Error Models for Object Recognition", Proc. CVPR, pp. 400-406.

3D Object Recognition Using Bidirectional Modular Networks

Hiroshi Ando

ATR Human Information Processing Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
ando@hip.atr.co.jp

Abstract. This paper investigates how the network scheme that consists of a set of bidirectional networks can be used to recognize gray-level images of 3D objects. The proposed scheme is based on the networks' ability to both compress and generate the input images. The paper also presents a bidirectional relaxation method that can be used to identify transformed or distorted images of 3D objects. We demonstrate through computer experiments that the proposed recognition model can learn to classify gray-level images of 3D objects and exhibit flexible alignment for transformed images.

1 Introduction

One of the important problems in 3D object recognition is how to recognize various 3D objects from their 2D projected images despite significant image variations as the viewpoint changes. Recently, computational models have been explored that learn to recognize 3D objects from their projected views [8, 6]. Most existing view-based models are, however, based on supervised learning, i.e., during training the teacher tells which object each view belongs to. Suzuki and Ando [9], on the other hand, proposed an unsupervised classification scheme for categorizing 3D objects from their 2D views. The scheme consists of a set of 5-layer autoassociative networks or bidirectional networks. The simulations using synthetic 3D wire-frame objects demonstrate that the model effectively learns to classify an input view as a 3D object.

This paper shows how the scheme that exploits a set of bidirectional networks can be used to recognize real gray-level images of 3D objects. The scheme does not require the identification and correspondence of features, because the scheme is based solely on the existence of non-linear mappings between a set of images of an object and its degree of freedom. The paper also presents a dynamic matching scheme based on bidirectional relaxation, which can identify a transformed or distorted image of a 3D object [4]. This scheme assumes an interaction between bottom-up and top-down information, i.e., it iterates between the generation of the image and the alignment of the input image with the generated image. It is also very likely that the human brain uses top-down information to guide the recognition of 3D objects [2]. The computer experiments demonstrate that the proposed model can effectively classify gray-level images of 3D objects and exhibit flexible alignment even when the input image suffers a substantial transformation.

2 The Network Model

This section describes a network model for classifying 2D views into 3D objects. Fig. 1 illustrates the overall scheme for 3D object recognition. We assume that the region of a 3D object is segmented in the image beforehand. The 2D projected image of the 3D object is first filtered by a smoothing function, such as Gaussian function or Gabor function. The filtered image is then sampled at grid points, each of which has a receptive field on the image. The filtered image on the grid is then fed into a set of modules which consist of 5-layer auto-associative networks or bidirectional networks. Each network is trained to compress the image nonlinearly and recover the image for a single object. The encoded information represents the view direction or the pose of the given 3D object. The bidirectional networks can also be trained in an unsupervised manner, i.e. no explicit object identity is needed to train the networks.

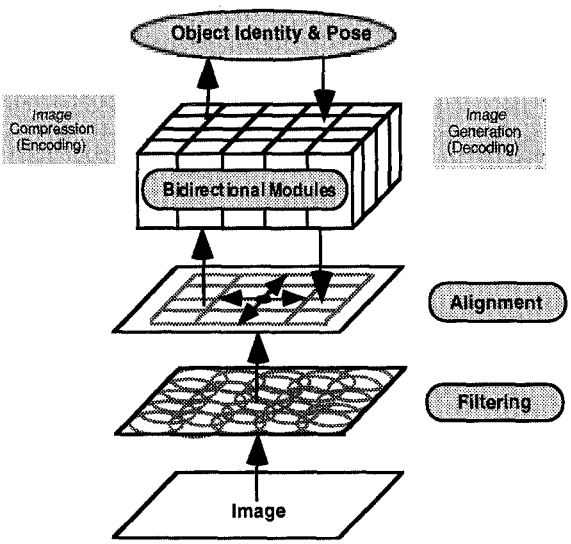


Fig. 1. The overall scheme for 3D object recognition.

After training the networks, an image of an object is given to the bidirectional modules. Each module encodes the image and generates its imagery. The generated images are then compared to the input image. The classifier identifies the input image as an object by selecting a module that best recovers the image. If the input image is distorted or transformed from the training views, the generated image does not match the input image in general. Even in such a case, each bidirectional network generates its most probable view because of its ability of associative recall (see Section 3.1). The system then performs alignment of the input and generated images by changing the position of each grid point. The aligned image is again provided to the bidirectional networks for verification. If the aligned image does not yet fit the newly generated image, the system performs alignment again. This bidirectional relaxation process is

iterated until the aligned image fits the generated image reasonably well. The following sub-sections describe the unsupervised learning method and the bidirectional relaxation process in more detail.

2.1 Unsupervised Learning of the Bidirectional Networks

In order to identify an object, the proposed scheme exploits an auto-associative network. An auto-associative network finds an identity mapping through a bottleneck in the hidden layer. The network, thus, compresses the input into a low dimensional representation by eliminating redundancy in the input set. If we use a five-layer perceptron network, the network can perform nonlinear dimensionality reduction, which is a nonlinear analogue to the principal component analysis [3, 7].

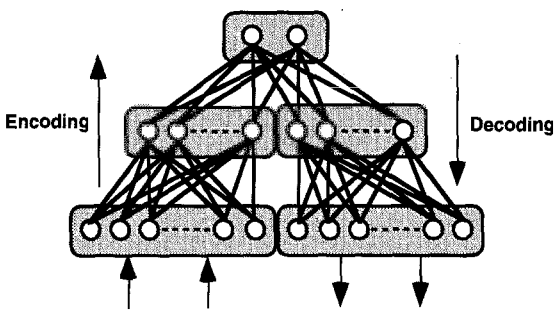


Fig. 2. Each module is a 5 layer auto-associative network.

The scheme consists of a set of five-layer neural networks, as illustrated in Fig. 2. In the case where the inputs are the projected views of a rigid object, the minimum dimension that constrains the input variation is *the degree of freedom* of the rigid object, which is six in the most general case, three for rotation and three for translation. Thus, a single module can compress the views of an object into a representation whose dimension is its degree of freedom. The proposed scheme categorizes each view of a number of 3D objects into its class through selecting an appropriate module.

The model selects a module whose output best fits the input. Specifically, we assume a classifier whose output vector is given by the softmax function of a negative squared difference between the input and the output of the modules, i.e.,

$$f_k = \exp\left[-\|I - R_k\|^2\right] \Big/ \sum_l \exp\left[-\|I - R_l\|^2\right] \tag{1}$$

where I and R_k denote the input and the output of the kth module, respectively. Therefore, if only one of the modules has an output that best matches the input, then the output value of the corresponding unit in the classifier becomes nearly one and the output values of the other units become nearly zero. For training the network with unsupervised learning, we maximize the following objective function:

$$\ln \frac{\sum_k \exp\left[-\alpha \|I - R_k\|^2\right]}{\sum_k \exp\left[-\|I - R_k\|^2\right]} \tag{2}$$

where α (>1) denotes a constant [10]. This function forces the output of at least one module to fit the input, and it also forces the rest of modules to increase the error between the input and the output. Since it is difficult for a single module to learn more than one object, we expect that the network will eventually converge to the state where each module identifies only one object.

2.2 Bidirectional Relaxation

When the input image suffers transformations or distortions, such as positional shift, scaling, image rotation or any nonrigid deformation, the image generated by a network may become significantly different from the input image. The underlying idea of the bidirectional relaxation is that the system allows each grid point or receptive field to move around the image until a resampled image best matches the image that the network generates.

There are several ways to move the grid points for aligning the input image and the generated image. One could use a global feature such as the center of gravity or the moment of the images for aligning the images. Alternatively, one could allow each grid point to move in different directions. Finding the best matched image in this case, however, is ill-posed, because each grid point can move to any image point that has the same gray value. One way to obtain a unique solution is to regularize the problem by imposing an additional constraint [9]. Our particular choice is a smoothness constraint which only allows a smooth topological distortion of the grid.

The scheme presented here is similar to the Dynamic Link Architecture (DLA), which is proposed for dealing with image distortions [1]. However, the DLA uses a fixed template for 2D pattern matching. The bidirectional scheme is, on the other hand, proposed for 3D object recognition so that it does not assume any fixed template for alignment. The scheme aligns the input image with the generated image which constantly changes as the grid points move.

3 Computer Experiments

We implemented the proposed network model to evaluate its performance on real images of 3D objects [4]. The 3D objects that we used for the experiments are three objects shown in Fig. 3. Each object was placed on a motorized turntable and different views of the objects were taken by a camera at every one degree of rotation. We thus obtained 360 images for each object. Each image has 512×480 pixels with 8 bit gray-scale values. We normalized intensities of the images for all objects by roughly setting the background intensity to 0.4 and the brightest portions to 1.6.

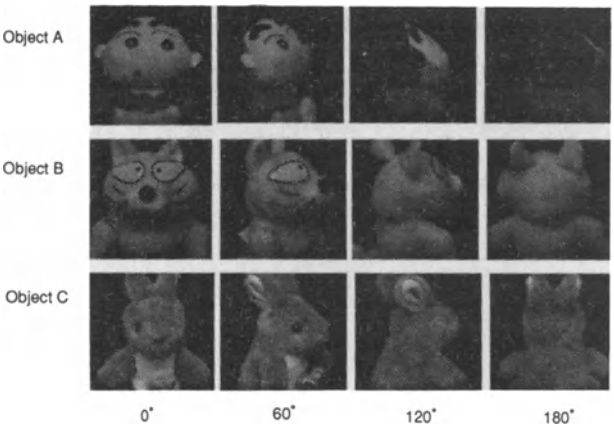


Fig. 3. Three objects used in the experiments: Object A (top), B (middle), and C (bottom).

3.1 Basic Properties of the Networks

This section examines basic properties of each module. For initial filtering and sampling, each image was quantized into a 32×30 image by averaging the brightness within each quantized cell. The input dimension of the networks is thus 960. 10 units were used in the second and fourth layers. The number of units in the third layer of the networks should be set equal to the degree of freedom, which is one in our experiments. However, we used two units instead because it is not possible to represent the periodic nature of rotation in just one dimension. To train each network, the squared difference between the input and output images of an object was minimized. At each iteration step, a view of the object was randomly selected from the entire views.

Fig. 4 and 5 show four set of examples of the input image and the recovered image for Object A and B after training the networks. The recovered views are significantly similar to the input views, indicating that each network can successfully compress and recover different views of an object. In fact, as shown in Fig. 6 (a) and (b), the reconstruction error between the input and the output images for two modules is nearly zero for only one of the three objects. Here, the reconstruction error is defined as a root mean squared error per pixel between the input and the output images. The results indicate that the classifier can correctly identify each image as an object by selecting a module that has the minimum reconstruction error.

To examine generalization performance of the model, we trained the networks with a sparse set of views, i.e. views are sampled only at 30 or 45 degree intervals. Fig. 6 (c) shows the reconstruction error of Object B plotted over the entire view range after training the networks. Comparison between figures (c) and (b) indicates that even when only 8 views with 45 degree intervals are used for training the network, the reconstruction error is smaller than the error of other objects for almost entire view range, which leads to correct classification of the objects. The results suggest that the network exhibits a satisfactory capability of generalization.

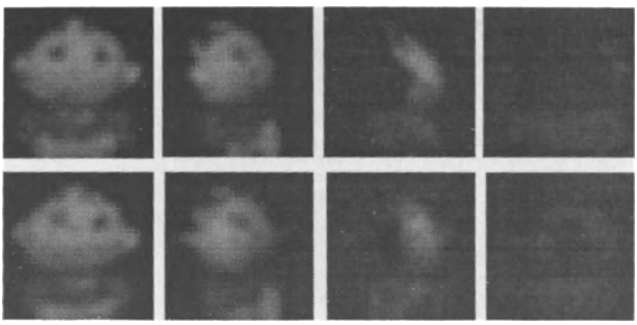


Fig. 4. Examples of the input view (top) and its recovered view (bottom) of Object A.

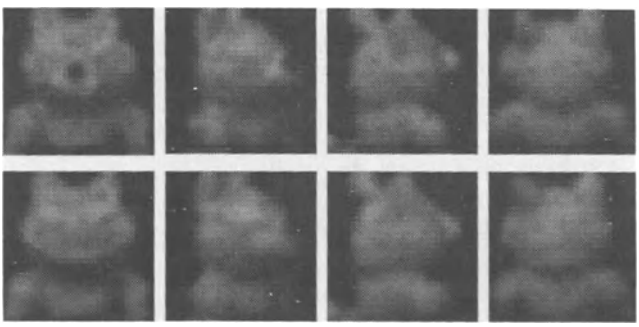


Fig. 5. Examples of the input view (top) and its recovered view (bottom) of Object B.

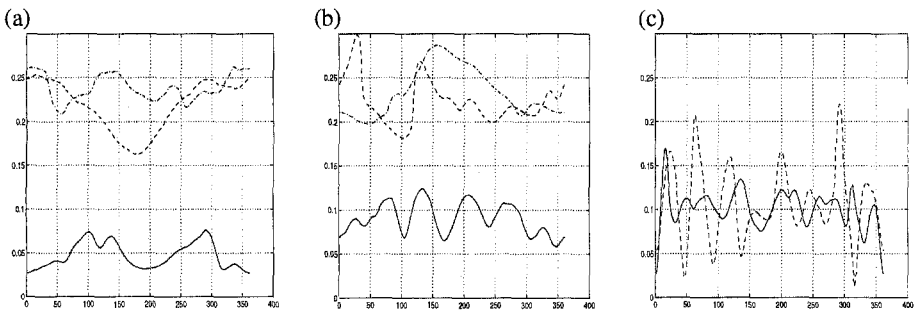


Fig. 6. (a), (b) The reconstruction error of two modules as a function of the view direction. The solid curves in (a) and (b) indicate Object A and B, respectively. Other curves indicate other objects. (c) The reconstruction error of Object B when the training views are sampled at 30 (the solid curve) or 45 (the dash curve) degree intervals.

We also analyzed the encoded information in the third-layer of the networks. Fig. 7 illustrates the outputs of the two hidden units in the third-layer of the network for Object B. The figure indicates that when the object is rotated, the outputs of the hidden units move along the shown curve. Thus, the representation acquired by the hidden units corresponds to the view direction or the pose of the object.

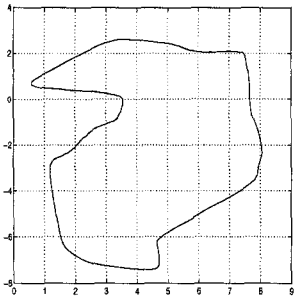


Fig. 7. Outputs of the two hidden units in the third-layer of the module for Object B.

One of the effective properties of a bidirectional network is its ability of associative recall [5]. The input image can be contaminated by noise, or a part of the object may be occluded by other objects in the image. We demonstrate how the network can reconstruct the original images for such noisy inputs. Fig. 8 (a) shows a case where a part of the image is masked. After learning the entire views, the network can effectively reproduce the original image from the masked image. By comparing the input pattern and the reconstructed pattern, the information of the missing portion can be estimated. Fig. 8 (b) demonstrates that the original image can also be recovered when the input image is superimposed with a white noise.

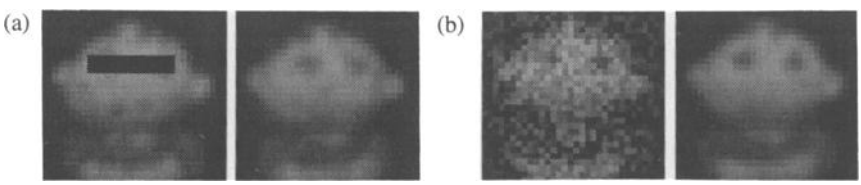


Fig. 8. Demonstration of auto-associative reconstruction in the case of the masked input pattern (a) and the noisy input pattern (b).

3.2 Unsupervised Classification

This section describes the experimental results using the unsupervised learning scheme described in Section 2.1. We used the same three objects used in the previous experiments, but object labels were not provided to the networks. 16×15 quantized images of the three objects were used for training the model which consists of three modules. The steepest ascent method was used for maximizing the objective function (2), but more efficient methods, such as the conjugate gradient method, can also be used. During the training, objects were randomly selected among the three and their views were also randomly selected. The constant α in the objective function (2) was set to 200. The classifier selects a network that produces minimum error between the output and the input. Fig. 8 illustrates the output values of the classifier plotted over the entire view direction after the training, showing that the model effectively classified the views into their original object except for the profile of Object C. The reason for

this misclassification is that the profile of Object C is very similar to that of Object B. We confirmed this by computing the squared difference between the images of Object B and C.

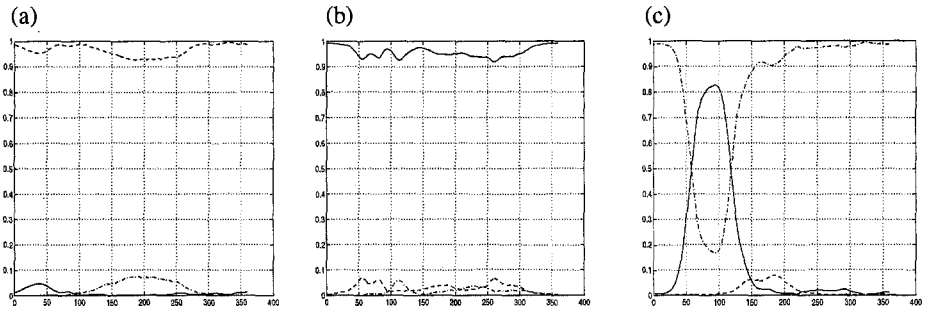


Fig. 8. The classifier outputs for the images of Object A (a), B (b) and C (c) as a function of the view direction. The solid, dash and dash-dot curves indicate the units corresponding to Module 1, 2 and 3, respectively.

3.3 Experiments on Bidirectional Relaxation

This section briefly describes the experimental results using the bidirectional relaxation [4]. The images were first blurred by Gaussian filter whose standard deviation was 4.0 pixels. After training the networks using the original images, a transformed image was used for the input. The left figure of Fig. 9 shows the transformed image of Object B sampled at 16×15 grid points. This image was shifted leftward and upward by 48 pixels each from its original image, and then further rotated by 10 degrees. Initially the error between the input and generated images was so large that the transformed image was misclassified as Object C. The bidirectional relaxation was then conducted using the combination of the center of gravity and the regularization method for the alignment (see Section 2.2).

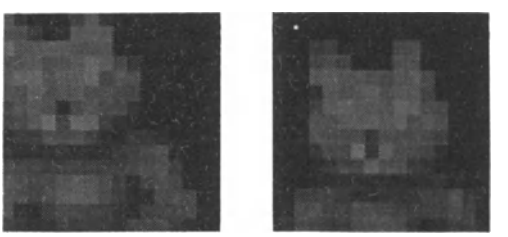


Fig. 9. The images sampled at the grid points before (left) and after the bidirectional relaxation (right).

The right figure of Fig. 9 illustrates the final state of the sampled image after the bidirectional relaxation. The figure shows that each grid point moved close to the original position so that the error between the sampled image and the generated image became very small. Thus, the object and its view were correctly estimated. These results demonstrate that the proposed bidirectional relaxation is effective even when the input image is largely transformed.

4 Conclusions

We have studied how the scheme that consists of a set of bidirectional networks can learn to recognize real gray-level images of 3D objects. We have developed a dynamic matching scheme, which can identify distorted images through bidirectional relaxation. The computer experiments demonstrated that the proposed model can effectively classify gray-level images of 3D objects and show a flexible alignment for a transformed image.

Acknowledgments

I would like to thank Mitsuo Kawato for continuous encouragement on this research.

References

1. Buhmann, J., Lange, J., von der Malsburg, C., Vorbrüggen, J.C. and Würtz, R.P. (1991). Object recognition in the dynamic link architecture: Parallel implementation of a transputer network. In B. Kosko (eds.) *Neural Networks for Signal Processing*, 121-159. Englewood Cliffs, NJ: Prentice Hall.
2. Cavanagh, P. (1991). What's up in top-down processing? In Gorea, A. (eds.) *Representations of vision: Trends and tacit assumptions in vision research*, 295-304. Cambridge University Press, Cambridge.
3. DeMers, D. and Cottrell, G. (1993). Non-linear dimensionality reduction. In Hanson, S. J., Cowan, J. D. & Giles, C. L., (eds), *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann Publishers, San Mateo, CA. 580-587.
4. Fujita, T., Suzuki, S. and Ando, H. (1995). A network model that recognizes 3D objects: Alignment based on regularization principle. *Technical Report of IEICE*, NC95 (in Japanese).
5. Kohonen, T. (1988). *Self-organization and associative memory*. 2nd ed. Springer-Verlag.
6. Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14, 5-24.
7. Oja, E. (1991). Data compression, feature extraction, and autoassociation in feedforward neural networks. In Kohonen, K. et al. (eds), *Artificial Neural Networks*. Elsevier Science publishers B.V., North-Holland.
8. Poggio, T. and Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature*, 343, 263.
9. Poggio, T., Torre, V. and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317, 314-319.
10. Suzuki, S. and Ando, H. (1995). Unsupervised classification of 3D Objects from 2D views. In Tesauro, G., Touretzky, D. & Leen, T. (eds), *Advances in Neural Information Processing Systems 7*. The MIT Press, Cambridge, MA., 949-956.

Outdoor Computer Vision

A Region Extension Method for Natural Outdoor Scene Interpretation

Michael Hild † and Yoshiaki Shirai
Dept. of Mechanical Engineering for Computer-Controlled Machinery, Faculty of Engineering
Osaka University, Yamada-oka 2-1, Suita 565, Japan
† Dept.of Engineering Informatics, Faculty of Information Science & Technology
Osaka Electro-Communication University, Hatsu-cho 18-8, Neyagawa, Osaka 572, Japan

Summary *This paper describes a method for natural scene interpretation which first extracts core regions of high reliability using local feature default models and qualitative constraints and then carries out final scene interpretation by extending these core regions to true object boundaries. Final scene interpretation proceeds by processing core regions with highest reliability first. For the determination of true object boundaries, maximal bounds for the object regions are determined using a hue depletion method and statistically robust curve fitting, followed by a step of compensating for local feature shifts based on hue impletion. This method allows the interpretation of difficult scenes in spite of local feature variations and weak object class separation.*

1 Introduction

The two main goals in scene interpretation are recognition of objects and segmentation of images into object regions. The major precondition for achieving reliable region segmentation requires that local features are approximately invariant within the object regions. The major precondition for reliable object recognition requires that local feature distributions are separable with regard to object classes. For natural outdoor scenes, typically neither of these conditions is satisfied. This paper addresses the problem of interpretation of scenes for which neither local feature invariance nor separation of local feature distributions is given.

Scene interpretation systems proposed in the past [1, 5, 6, 2, 8, 4] usually compute a single initial segmentation of an image in a data-driven manner, solely based on similarity of local features as the first step and rely only on a fixed set of features for extracting objects. A fixed feature set, however, is too inflexible for coping with the variability of natural outdoor scenes. Strat's system [8] appears to be more robust against local feature variations due to selecting the largest consistent set of hypotheses based on scene context.

This paper's approach to coping with the two problems of *feature variability* and *weak class separation* consists of extracting those regions first which are segmentable and immediately recognizable with sufficient reliability (called core

regions) in a first stage, and later try to extend them up to their real object boundaries in a second stage. Core regions play the role of indicators of the approximate position, size and orientation of object regions, but unlike genuine object regions, core regions need not be *complete*. Core regions focus the attention of the system on most relevant image areas. Our method employs *multiple local features* (including texture) and *multiple parameter settings* for core region extraction. The extraction mechanism computes *multiple segmentations* of objects of the same class and extracts only regions satisfying object constraints (shape, location, orientation, spatial relationships). Multiple, spatially overlapping regions obtained from the same local feature are combined, and regions from multiple local features are integrated to form core regions for a given object class. Core regions are assigned a reliability measure used in the second stage to refine the results of the first stage in the order of highest reliabilty. Beginning at core regions, constrained searches for object boundaries are performed in restricted image areas and results from all object classes are integrated in order to achieve a consistent scene interpretation.

2 First stage: Core Region Extraction

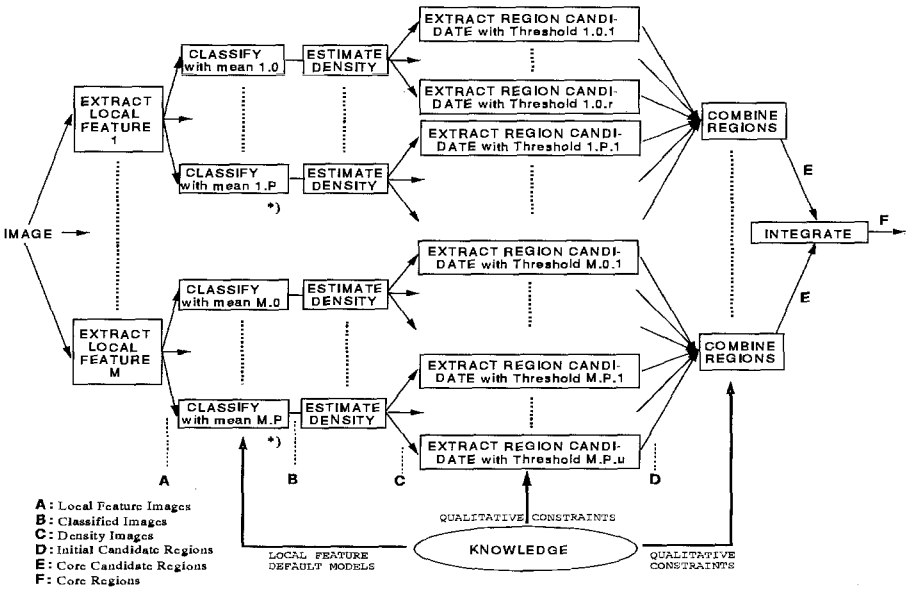
Knowledge about scene objects usually includes global features (e.g. shape, size and orientation of regions) and local features (e.g. likelihood functions of surface color, texture coarseness, etc.).

For natural outdoor scenes, it is usually impossible to provide an exact description of the objects' global features since global features often have significant variance within the same class. Global knowledge is therefore represented as *qualitative constraints* on global properties of scene objects.

Local features, too, vary a great deal in natural outdoor scenes. Local Feature Default Models (LFDM) are introduced to cope with this problem. LFDM consist of *default distributions* for the values of the local features and a *valid range for the mean values* of such distributions. The LFDM is more flexible than a fixed distribution would be, because it allows the adjustment of the mean distribution values within the valid range [3].

An overview of the core region extraction stage is shown in Figure 1. First, for a target object class and for every meaningful local feature, the pixels of feature images (A) are assigned to candidate object classes by using the local feature default models for the object classes. One such classification is computed for each distribution mean value specified by the default model of the target object class. The result is a set of classified target object images (B). A density image (C) is estimated from each classified target object image which is then thresholded with a set of threshold values that are determined individually for each image on the basis of the image's robust statistics. [7] This step yields a set of region images for each feature containing raw regions. Each of these raw regions is examined to determine whether the regions or some region parts satisfy qualitative spatial constraints imposed by the context for the target object class. If such regions or region parts are found, they are extracted and called initial candidate

regions (D). Overlapping initial candidate regions are then combined, subject to constraints, to form the candidate core region of the given feature channel (E). Candidate core regions from different feature channels are integrated to form the core regions for the target object class (F). One of these core region extraction mechanisms is provided for each object class in the scene.



*) Note that the number of distribution mean values may be different for different features.

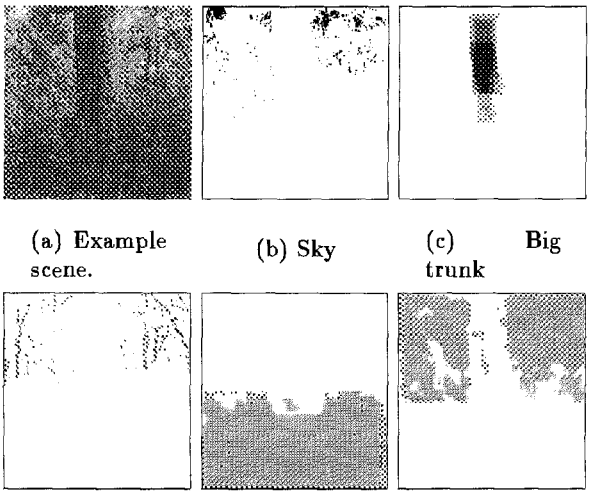
Fig. 1. Overview of candidate extraction by evaluation of regions obtained through multi-parameter process

As is obvious from Figure 1, the structure of this process allows a great deal of parallel processing. There exist *parallel feature channels*, and within these channels there are *parallel classifiers*; the classification results are fed to *parallel region candidate extractors.*

Figure 2 shows an example scene and the core regions that were extracted for the object classes tree trunk, grass, and tree leaves. Details of this stage may be found in [3].

3 Second stage: Core region-based interpretation of entire scenes

The sets of core regions generated by the first stage serve as the starting points for the final image interpretation process. With regard to the processing requirements of the second stage, two points have to be taken into consideration.



(a) Example scene. (b) Sky (c) Big trunk

(d) Tree branches (e) Grass (f) Tree leaves

Fig. 2. Core regions for all object classes of Scene 1: Grayvalue encodes the reliability measure assigned to the core regions

Firstly, since extracted core regions are only fragmentary cores suitably representing object regions, the determination of true object region boundaries is still required. Secondly, extracted core regions from different object classes may overlap spatially, leading to conflicts. Final interpretation of the image taking into consideration these two points is realized by the following outline algorithm.

Final interpretation algorithm (outline):

1. Extend the core region having highest reliability to the supposedly true boundaries of the objects.
2. During the process of extending regions with highest reliability, eliminate less reliable regions (or region parts) if they are completely (or partly) overlapped by the regions being extended.
3. If unextended core regions remain, go to step (1). If not, exit.

For region extension, two cases must be distinguished according to whether shape constraints are available or not. When shape constraints are available, it is usually possible to extract rather accurate region boundaries because these constraints allow the initialization of knowledge-guided model fitting methods. When shape constraints are unavailable, only the regions of other objects may be used as – rather weak – constraints.

3.1 Case 1: Shape constraints are available

The determination of true object region boundaries is usually possible if a set of several different features is used; in this section, only the method for the important feature "hue" is described. The main point is that hue shifts, which are mainly caused by shading in the scene, are compensated before exact boundary detection takes place.

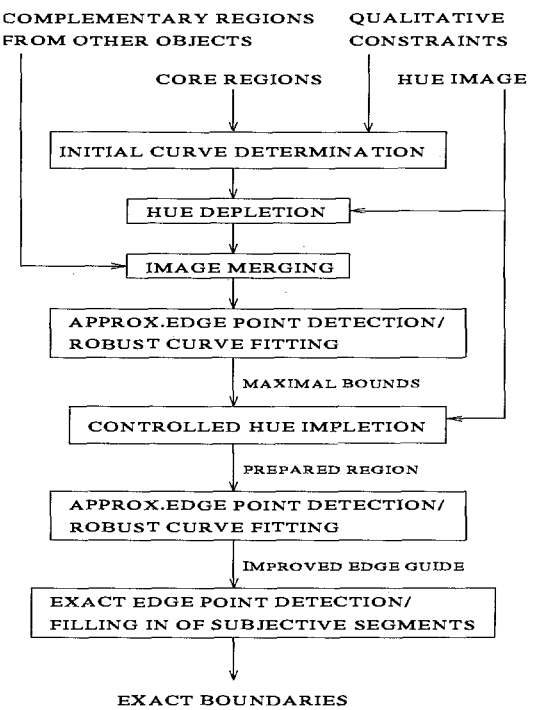


Fig. 3. Overview of method for precise boundary determination

Figure 3 shows an overview of the procedure for precise boundary detection. This procedure solves the problem by attaining three consecutive sub-goals: 1) The procedure tries to obtain *maximal bounds* for the object's boundary location; the start line is on a core region. 2) With the knowledge of these maximal bounds, the feature shift across the object region is estimated and a *hue shift compensated region* is prepared to obtains an improved estimate of the boundary location. 3) The *exact boundary* is then determined in the vicinity of the estimated boundary location. This procedure embodies a cautious coarse-to-fine approach to region boundary detection which takes into account known object regions of other object classes and tries to avoid losing track of the boundary's global properties.

Determination of maximal bounds on the boundary location First, the approximate location of a candidate region's true object boundary are determined. This search starts on a known core region using the following algorithm:

Maximal bounds on boundary location

1. Select an *initial curve* on a candidate region and measure the average hue (H0) on this initial curve.
2. Determine the *direction of hue shift* occuring between initial curve and boundary by comparing the average hue on the initial curve with the average hue on an assumed curve parallel to the initial curve, located between initial curve and boundary.
3. *Deplete* all *hues* in the interval [H0,H0+ΔH] from the image (ΔH extends in the direction of the hue shift; $0° < \Delta H < 180°$ holds, because hue shifts occur only up to the complementary hue; the value of ΔH is assumed by experience).
4. Use an oriented, noise-tolerant, directed edge detector (Figure 5a) to find edge point candidates of the boundary. Initially, it is placed on each point of the initial curve, oriented roughly at a right angle to the initial curve's tangent.
5. Fit a curve (called "edge guide") to the edge points using a robust model fitting technique [7] in order to distinguish between true and false (outlier) boundary points. The curve's shape is initialized according to known qualitative constraints.

Since some of the extracted points may not be true boundary points, they are eliminated using a statistically robust, 1D curve fitting procedure which identifies outliers in the distances between edge points and initial curve based on the LMS upper and lower outlier bounds given by Eqs. (1) and (2):

$$B_u = \min(d_{95\%}, \bar{d}_{lms} + k \cdot \sigma_{lms}) \tag{1}$$

$$B_l = \max(d_{5\%}, \bar{d}_{lms} - k \cdot \sigma_{lms}) \tag{2}$$

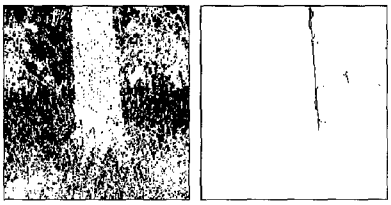
$\bar{d}_{lms}$ is the LMS (Least Median of Squares) [7] estimate, σ_{lms} is the robust standard deviation of the data, and $d_{5\%}$ and $d_{95\%}$ are the dodecile values of the ordered data set. Only inlier edge points are used for curve fitting based on a standard, inexpensive least squares procedure.

The hue depleted image for the tree trunk of Scene 1 (Figure 2a), edge points, and the edge guide for the right-hand trunk boundary are shown in Figure 4.

Improved boundary through feature shift compensation After the edge guide has become known, a hue shift compensated region is prepared for exact edge point detection. The amount hue shift up to the boundary is estimated through the execution a *hue impletion loop.*

Hue impletion loop:

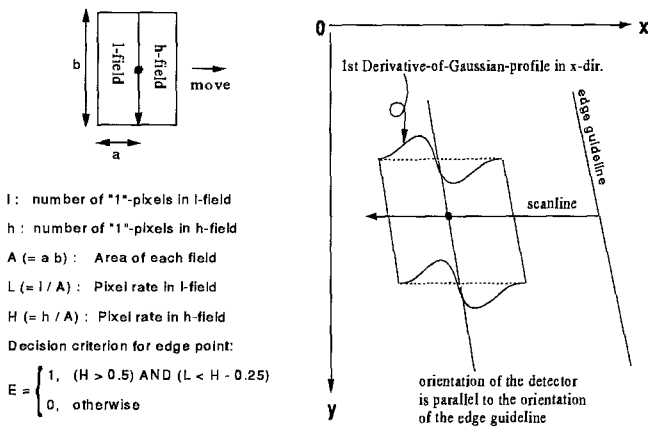
1. Select a hue interval [H0,H0+h], where H0 is the average hue on the initial curve and h is the magnitude of a slight shift in the known direction.



(a) Merged image

(b) Edge points and fitted guides.

Fig. 4. Maximal bounds on boundary location



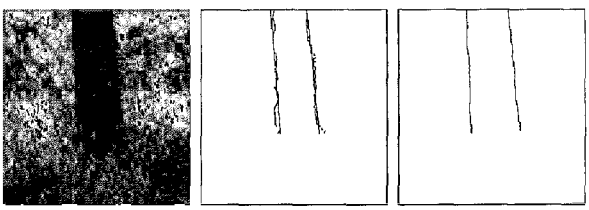
(a) Directed noise-tolerant EPD

(b) 1st Derivative-of-Gaussian EPD

Fig. 5. Edge point detectors

2. Prepare an image of only those pixels whose hue falls into the interval [H0,H0+h].
3. Measure the pixel density along the edge guideline in this image.
4. If this pixel density does not reached a sufficiently high value, widen the interval by some amount in the direction of the hue shift and go to step (2).
5. If the density has reached a sufficiently high value, or if the density has converged to some not too small value, stop the iteration and accept the present width h of the hue interval as the magnitude of the hue shift across the region.

On the basis of the hue impleted image, an improved edge guide is computed by using the appropriately oriented, noise-tolerant edge detector and fitting a model curve to only the distance inlier points (analogously to method in preceding sub-section). The results of this procedure is shown in Figure 6.



(a) Hue impletion result (b) Edge points/guides (c) Extracted trunk boundaries

Fig. 6. Improved edge guides

Exact determination of boundary points The improved edge guide is used as reference for a precise boundary detection operation applied to the prepared region in which a majority of edge points is detected and exactly localized by using a 2D First-Derivative-of-Gaussian operator, whose orientation is determined by the edge guide (Figure 5b), and by subsequently applied maximum (minimum) detection in the neighborhood of the edge guide. Because the operator is constrained to the neighborhood of the edge guide and the given orientation, only a few false edge points will occur; they are detected and eliminated as distance outliers with the LMS method. Since edge points which are identified as distance outliers, gaps in the detected boundary may occur; these gaps are treated as *subjective boundary segments* and are filled in with appropriate straight line segments, giving the result shown in Figure 6c.

3.2 Case 2: Shape constraints are unavailable

In contrast to strong qualitative constraints such as "shape", weak qualitative constraints such as region "position" and region "orientation" cannot be directly used for finding the true boundaries of object regions. If, in addition, true boundaries are hardly distinguishable on the basis of local image features, then true region boundary detection becomes an extremely difficult task. Here, the missing strong qualitative constraints may be supplemented by *occupancy constraints* provided by regions from more reliable object classes to estimate bounds for region extension. When the used feature is hue, hue impletion analogous to the

one described in sub-section 3.1 can be utilized for extending a region, but unlike that method, exact localization of edge points with the aid of 1st-Derivative-of-Gaussian detectors does not make sense here, because the employed constraints are too weak. Results for the grass region are shown in Figure 7.

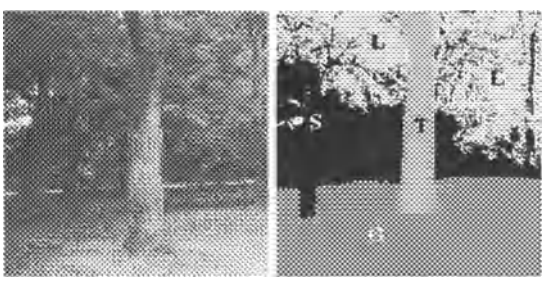
(a) Already determined regions

(b) Hue-impleted image

(c) Grass region

Fig. 7. Grass boundary

Figure 8 shows one more example scene and its interpretation result.



(a) Scene

(b) Interpretation result (G: ground; T: trunk; L: leaves; S: sky)

Fig. 8. Second example scene

4 Conclusion

This paper proposes a system for interpreting images of natural outdoor scenes where local features are highly variable across object regions and local feature distributions are not well separated. The main contribution of this paper consists of a compact, systematic and highly parallel method for knowledge-based interpretation of such scenes. It employs multiple local features and multiple parameter settings to derive multiple segmentations of an image and extracts core regions using local feature default models and qualitative constraints. Core regions are then extended in the order of highest reliability.

Although the fundamental structure of a system capable of interpreting complex natural outdoor scenes has emerged, several areas need continuing attention. E.g., the method for extending core regions to true object boundaries should be diversified to include multiple local features, and a richer set of qualitative constraints is desirable. Progress in these and other directions would contribute to making the system a more realistic, autonomous entity.

References

1. H. G. Barrow and J. M. Tenenbaum, Recovering Intrinsic Scene Characteristics From Images, Tech. Note 157, SRI International, Menlo Park (1978).
2. A. Hanson and E. Riseman, The Visions Image-Understanding System, in: C. Brown (ed.), Advances in Computer Vision, Vol.1, Lawrence Erlbaum Associates, New Jersey (1988).
3. M. Hild and Y. Shirai, Interpretation of Natural Scenes Using Multi-Parameter Default Models and Qualitative Constraints, Proc. of the Fourth Int. Conf. on Computer Vision, Berlin, 497-501 (1993).
4. S. Hirata, Y. Shirai and M. Asada, Scene Interpretation Using 3-D Information Extracted from Monocular Color Images, Proc. First Korea-Japan Joint Conf. on Computer Vision, 81-87 (1991).
5. M. Nagao, T. Matsuyama and Y. Ikeda, Region Extraction and Shape Analysis of Aerial Photographs, Proc. 4th Intl. Joint Conf. on Pattern Recognition, Kyoto, 620-628 (1978).
6. Y. Ohta, Knowledge-based Interpretation of Outdoor Natural Scenes, Pitman Publ. Inc., Marshfield, Mass. (1985).
7. P. J. Rousseeuw and A.M. Leroy, Robust Regression and Outlier Detection, John Wiley & Sons, New York (1987).
8. T. M. Strat and A. Fischler, Context-Based Vision: Recognizing Objects Using Information from Both 2-D and 3-D Imagery, IEEE Trans. PAMI, Vol.13, No.10, 1050-1065 (1991).

Image Processing Techniques for Crowd Density Estimation Using a Reference Image

Jia Hong Yin, Sergio A. Velastin and Anthony C. Davies

Vision and Robotics Laboratory
Department of Electronic and Electrical Engineering
King's College London, Strand, London WC2R 2LS, UK
j.yin@kcl.ac.uk, s.velastin@kcl.ac.uk, tonydavies@kcl.ac.uk

Abstract. There are still significant problems in the planning, design and management of public facilities subject to dense pedestrian traffic. The automation of data collection and analysis of crowd behaviour is increasingly desirable in design of facilities and long-term site management using image processing techniques with existing closed-circuit television systems. We have investigated a number of techniques for crowd density estimation, movement estimation, incident detection and their relative merits using image processing. This paper presents techniques for background generation and calibration to enhance the previously-developed method of crowd density estimation using a reference image. An intensity region related to the average pixel intensity of each image in a sequence of crowd images is used to segment background pixels for generating a background image without pedestrians. The calibration approach, with which a previously-established relationship between image parameters and crowd density at one site can be used to estimate crowd density at various sites, involves calibration of the crowd image as opposed to calibration of the camera. Both techniques may be used in other surveillance systems such as vehicle monitoring.

1. Introduction

Crowd data such as density and flow are an important factor in the planning, design and management of public facilities subject to dense pedestrian traffic. It is well-known that conventional manual measurement techniques are not suitable for comprehensive data collection of patterns of site occupation and movement in such situations and automation of data collection to assist in understanding crowd behaviour is highly desirable [1]. Previous studies [2, 3] have shown the feasibility of crowd data estimation using image processing techniques with existing closed-circuit television systems. This may be applied to improving design of the built environment and increasing pedestrian safety.

Our initial procedure for crowd density estimation is to establish relationships between image parameters from various image processing techniques and actual crowd densities at an investigated site [4]. One of the techniques developed is to classify the pixels occupied by pedestrians in crowd images by comparison with a reference image with only background. Clearly, the reference image quality in this method influences the accuracy of measurement. The direct way to obtain a

background image is to capture one from the investigated site when no pedestrians are present. However, in typical public areas, there are almost always some pedestrians present, and direct acquisition of an image with background only is often impracticable. Furthermore, inaccurate estimation may result from using the manually captured reference image because of effects of subsequent change in lighting level. Therefore, it is desirable to have an automatic method to generate reference images with only background for crowd density measurement at any time or at various investigated sites.

Another problem of these methods is that the establishment of a relationship between image parameters and number of people has to be repeated for each new site, which is time consuming and seriously restricts the adoption of automated methods. For general applicability of the developed methods, a faster method is needed which can transfer the previously-established image parameter to crowd density relationship in one investigated site to any other site.

In this paper, the technique of using a reference image to estimate crowd density is reviewed first. This is followed by introducing an automatic method to generate a background image from a sequence of crowd images. A calibration approach, by which the established relationship between image parameters and crowd density at one site can be used to estimate crowd density at other sites or for other camera positions, is presented with experimental results.

2. Density Estimation Using a Reference Image

The hypothesis of our method of density estimation using a reference image is that the area occupied by pedestrians in a crowd image contains information about crowd density in the scene. A fast method has been developed to estimate crowd density using a reference image with only background to classify pixels in a crowd image by subtraction as belonging to either pedestrians or background. Figure 2 shows a result image from Figure 1 which is a typical crowd image from one investigated site called "Site1".

Fig. 1: Image from "Site1"

Fig. 2: Result image from Fig. 1

By comparing the number of pedestrian-classified pixels and number of people counted manually, a clear tendency is found out that the number of pixels increases approximately linearly with increase of number of pedestrians, as shown in Figure 3. A linear function relating number of pedestrian-classified pixels (N_{px}) to number of people (N_p) is obtained by using a least squares fit, as shown as follows:

$$N_p = a N_{px} + b \tag{1}$$

where $a=0.00047$ and $b=-2.81$ for "Site1". The standard deviation is 1.1 pedestrians. This linear function has been shown adequate for moderate crowd densities (e.g. up to 1.25 pedestrians/m^2 [5]).

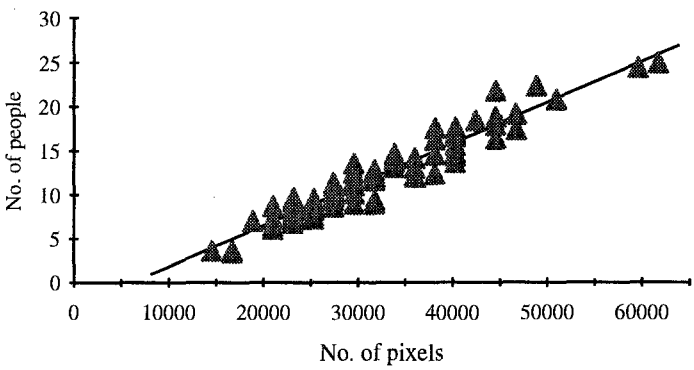


Fig. 3: No. of people vs. No. of classified pixels

3. Background Generation

The technique using a reference image to classify pedestrian pixels has clearly shown its feasibility for crowd density measurement. However, the applicability and accuracy of this method obviously relies on the reference image. Apart from the inconvenience of capturing such a reference image without pedestrians in public facilities, accuracy deteriorates as a result of subsequent changes in lighting level and direction. Therefore, an automatic method to generate background images for this density estimation method is desirable.

Rodriguez [6] has obtained good results on image segmentation by successive background extraction, but he required local homogeneous background images, and he did not study the removal of objects from background. The contributions from Long [7] include *Smoothness Detector Method*, *Adaptive Smoothness Detector Method* and *Stable Fill Method* to detect background using a sequence of images with a moving object. The first produced good results in some views but limitations occur when the object was large and moved slowly. Many iterations to process the sequence are used in the latter two methods, which is time consuming and is not practical in surveillance systems. Our approach, to be described, aims to automatically generate background images from crowd video sequences, which are used as reference images for crowd density measurement.

3.1 Methodology

The procedure for background generation is shown in Figure 4. A region with intensities between $\overline{A}-\Re$ and $\overline{A}$ is selected and used to classify pixels as background in each image of the video sequence. $\overline{A}$ denotes the average pixel intensity in each image, and $\Re$ is related to intensity variation of background pixels and is found by trial. Pixels in each image with intensities between $\overline{A}-\Re$ and $\overline{A}$ are classified as background pixels and extracted into a background frame store. This process is repeated for each image in the sequence, exploiting the fact that as people move in the sequence, more portions of the background are exposed so that for those parts where background pixels cannot be obtained from the first image they may be captured from the next. Any parts whose background pixels cannot be captured throughout the whole sequence are allocated an intensity corresponding to their value in the first crowd image. This assumes that these unclassified parts must belong to a part of the image which does not change throughout the whole sequence.

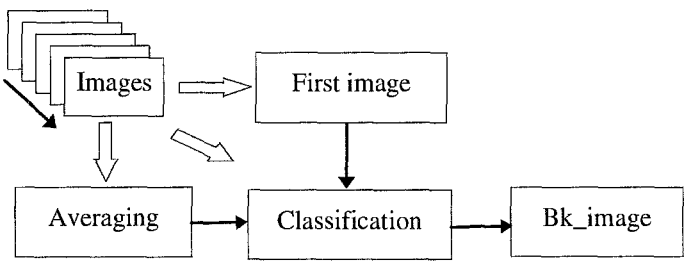


Fig. 4: Background generation procedure

3.2 Analysis of Results

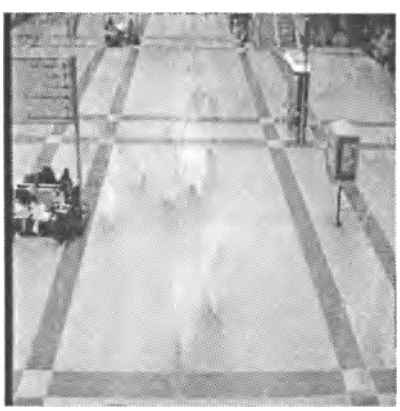


Fig. 5: Generated background image

This method has been used to generate background images in our investigated sites for use as reference images to estimate crowd density in both data gathering and

"incident" detection. One generated background image for "Site1" is shown in Figure 5. The results from this approach are adequate for use in our applications. However, results may be poor if the intensity of pedestrian clothing is similar to the background. Also for those parts of the image where pedestrians never move throughout the sequence, the background cannot be recovered.

The generated background image shown in Figure 5 has been used to estimate crowd density at "Site1". Results using both a generated image and a reference image taken from the site have been compared with manual counts. The distribution of the *Absolute Error* (defined as the absolute difference between automatic counts and manual counts for each image) is shown in Figure 6. Figure 7 shows the *Maximum Relative Error* (defined as the ratio of maximum absolute error to the manual count). The results from using the generated reference image are very close to those using a reference image taken from the site.

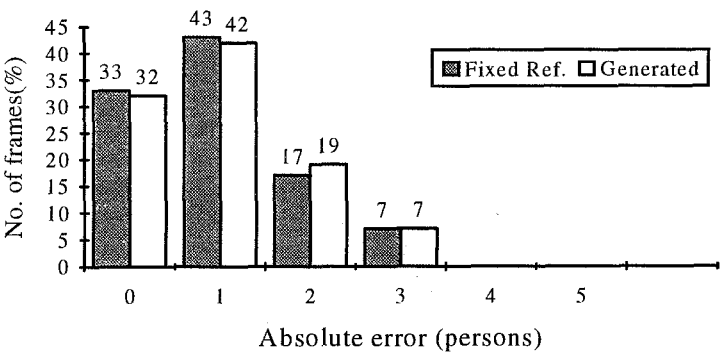


Fig. 6: Absolute error

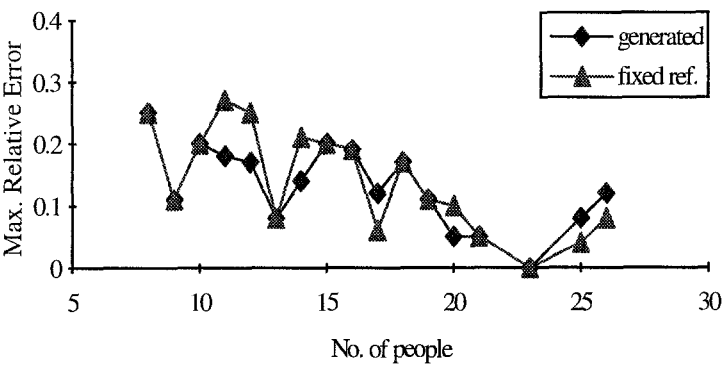


Fig. 7: Maximum relative error

4. Calibration for Density Measurement at Various Sites

Equation (1) can be used to measure crowd density but only for the site and the camera position at which the slope a and intercept b are obtained. Using this function to estimate crowd density at other sites (or other camera positions) is possible only if a relationship between the parameters from "Site1" and those applicable to each other site or camera position can be established. Some existing techniques in 3-D computer vision for camera calibration involve computing intrinsic and extrinsic camera parameters based on knowledge of 3-D coordinates, such as using a number of points [8, 9] or various targets [10], and their corresponding 2-D image coordinates. However, to avoid the complexity inherent in such methods, the calibration approach proposed here is intended to calibrate the crowd image as opposed to calibrating camera parameters. It exploits environmental constraints with which the relationship established at "Site1" can be used to estimate crowd density for other camera positions or for other sites.

4.1 Formulation

Using a conventional simple camera model and from the geometry of two similar triangles, the height h of an object in an image plane is approximately linearly proportional to its height H in 3-D world, and linearly proportional to its image position x as the object moves within the camera view [11]. If two objects of heights H_1 and H_2 placed at the same point are viewed using camera position one, and their corresponding heights in the image plane are h_1 and h_2, then $h_1 / h_2 = H_1 / H_2$. If the two objects are then viewed from a different camera position, called position two, then $h_1^* / h_2^* = H_1 / H_2$ where * denotes their heights in the second image plane. Therefore the heights in the two image planes are directly related as follows:

$$h_1 \cdot h_2^* = h_1^* \cdot h_2 \tag{2}$$

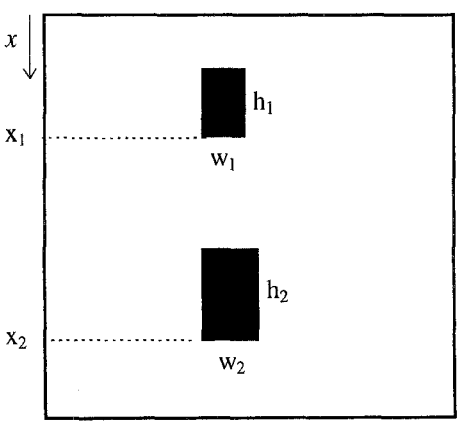


Fig. 8: Image for calibration

Crowd density is measured in terms of "nominal" (average) pedestrians. For the purpose of calibration, a nominal pedestrian can be described by circumscribing rectangles, located near and away from the camera (Figure 8) both in the known image ("Site1") and the image to be calibrated ("Site2"). The process can be based on images acquired using calibrating objects. When this is not possible, the nominal rectangles can be estimated manually.

Suppose that the rear rectangle at x_1 in the image plane of "Site1" has height h_1 and width w_1 and the front rectangle at x_2 has height h_2 and width w_2. The following equations can be derived from the linear relationship between image height h and its position x:

$$h = A_{Ih}x + B_{Ih} \tag{3}$$

where $A_{Ih} = \dfrac{h_2 - h_1}{x_2 - x_1}$ and $B_{Ih} = \dfrac{h_1 x_2 - h_2 x_1}{x_2 - x_1}$.

$$w = A_{Iw}x + B_{Iw} \tag{4}$$

where $A_{Iw} = \dfrac{w_2 - w_1}{x_2 - x_1}$ and $B_{Iw} = \dfrac{w_1 x_2 - w_2 x_1}{x_2 - x_1}$.

Let $s_I(x)$ indicate the area $h \cdot w$ of the rectangle in the image plane of "Site1", which can be described by equation (5):

$$s_I(x) = S_{I1}x^2 + S_{I2}x + S_{I3} \tag{5}$$

where $S_{I1} = A_{Ih}A_{Iw}$, $S_{I2} = A_{Ih}B_{Iw} + A_{Iw}B_{Ih}$ and $S_{I3} = B_{Ih}B_{Iw}$.

Similarly the area $s_{II}(x)$ of the rectangle in image plane of "Site2" is given by:

$$s_{II}(x) = S_{II1}x^2 + S_{II2}x + S_{II3} \tag{6}$$

Based on the concept of nominal pedestrian size, if the number of pedestrian-classified pixels at x in a "Site2" image is $N_{II}(x)$, the number of corresponding pixels at x, $N_I(x)$, in a "Site1" image can be calculated by equation (7) in accordance with equation (2).

$$N_I(x) = \frac{s_I(x)}{s_{II}(x)} N_{II}(x) = \delta(x)\, N_{II}(x) \tag{7}$$

where $\delta(x) = \dfrac{S_{I1}x^2 + S_{I2}x + S_{I3}}{S_{II1}x^2 + S_{II2}x + S_{II3}}$.

The total number of pedestrian-classified pixels in an area of interest (AOI) in the "Site2" image can be equivalently converted to the total number of pixels in the "Site1" image with the following equation:

$$N_{px} = \sum_{x \in AOI} \delta(x) N_{II}(x) \tag{8}$$

which can be substituted into equation (1) to calculate the number of people in the area of interest in the "Site2" image.

The procedure of estimating crowd density using this calibration approach involves the following steps: $\delta(x)$ is first obtained by using the nominal rectangle located in "Site1" (the camera position where the function between number of

pedestrian-classified pixels and number of people is known) and in "Site2" (where the measurement of crowd density is required). Then, the number of corresponding pedestrian-classified pixels is calculated using equation (8). Finally this number is converted to number of people at the investigated site using equation (1).

4.2 Experimental Results

Fig. 9: Image from "Site2"

Fig. 10: Image from "Site3"

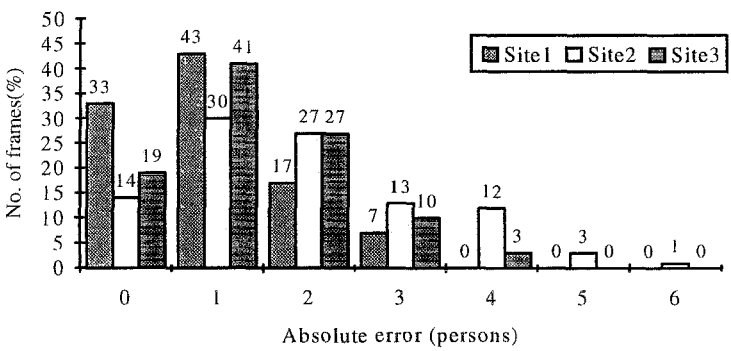


Fig. 11: Absolute error

This calibration approach has been used experimentally to estimate crowd density in two sites, called "Site2" and "Site3", together with equation (1) obtained from "Site1". $\delta(x)$ was obtained by measuring the area covering pedestrians in the images. Two background images were generated using the automatic method for the two sites. At "Site2", shown in Figure 9, large crowds of people had built up waiting for trains. Figure 10 shows a typical image from "Site3", a view in front of escalators. The results from the two sites have been analysed by comparing those obtained from "Site1" using a reference image taken directly in the site. Figure 11 and Figure 12 show the comparison of *Absolute Error* and *Maximum Relative Error* respectively.

Results from "Site2" have errors larger than those for "Site3" due to over-crowding and poor lighting condition in the evening. The results from both sites are good enough for the purposes of automatic crowd monitoring and incident detection.

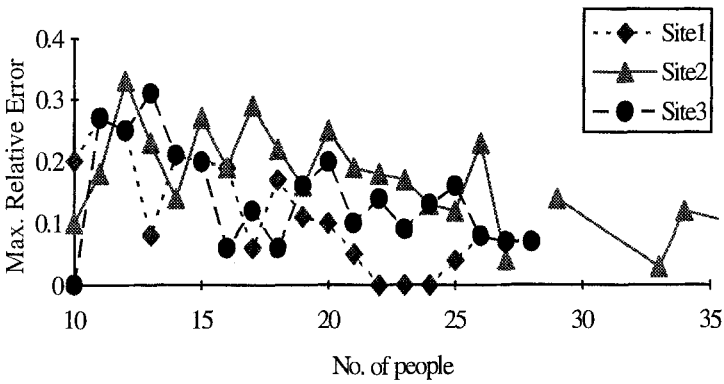


Fig. 12: Maximum relative error

5. Conclusions

This paper has presented two image processing techniques to enhance crowd density measurement using a reference image. A reference image with only background is used to classify image pixels as belonging to either pedestrians or background so that a functional relationship between number of pedestrian-classified pixels and number of people can be established for crowd density measurement. The automatic background generator proposed exploits an intensity region related to the average pixel intensity of each image to segment background pixels and puts background pixels from a sequence of crowd images together to obtain an artificial background image. The calibration approach, with which a previously-established function for one site can be used to estimate crowd density at various other sites, involves calibration of the image as opposed to calibration of the camera. The results from two investigated sites using the background generator and the calibration have been shown to be compatible with those from the site where the function between number of pedestrian-classified pixels and number of people was established with a reference image taken from site. The techniques increase the general applicability of the crowd-density estimation method developed by the authors. Both background generation and calibration may be used in other surveillance systems such as vehicle monitoring.

6. Acknowledgements

The authors are grateful to the UK EPSRC for its financial support (grants GR/H78511 and GR/83539), to Railtrack East Anglia (UK) for granting access to their sites, and to their colleagues X. Zhang, M. A. Vicencio-Silva, R. E. Allsop, L. Vaughan and A. Penn (University College London) for their collaboration.

7. References

[1] Davies A.C., Yin J.H. and Velastin S.A. (1995): "Crowd monitoring using image processing", IEE Electronics Communication Engineering Journal, Vol. 7, No. 1, February, pp 37-47.

[2] Velastin S.A., Yin J.H., Davies A.C., Vicencio-Silva M.A., Allsop R.E. and Penn A. (1994): "Automatic Measurement of Crowd Density and Motion Using Image Processing", IEE 7th International Conference on Road Traffic Monitoring and Control, 26-28 April 1994, London, UK, pp 127-132.

[3] Velastin S.A., Yin J.H., Davies A.C., Vicencio-Silva M.A., Allsop R.E. and Penn A. (1994): "Image Processing for On-line Analysis of Crowds in Public Areas", 7th IFAC/IFORS Symposium on Transportation Systems: Theory and Application of Advanced Technology. 24-26 August, Tianjin, China, pp 170-175.

[4] Yin J.H., Velastin S.A. and Davies A.C. (1994): "Measurement of Crowd Density Using Image Processing", EUSIPCO'94, September, Edinburgh, UK, pp 1397-1400.

[5] Polus A., Schofer J. L. and Ushpiz A. (1983): "Pedestrian Flow and Level of Service", Journal of Transportation Engineering, Vol. 109, No. 1, January, pp 46-56.

[6] Rodriguez A. A. and Mitchell O. R. (1991): "Image Segmentation by Successive Background Extraction", Pattern Recognition, Vol. 24, No. 5, pp 409-420.

[7] Long W. and Yang Y. (1990): "Stationary Background Generation: An Alternative to the Difference of Two Images", Pattern Recognition, Vol. 23, No. 12, pp 1351-1359.

[8] Tsai R.Y. (1987): "A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August, pp 323-344.

[9] Wei G.Q., Ma S.D. (1994): "Implicit and Explicit Camera Calibration: Theory and Experiments", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 5, May, pp 469-480.

[10] Wang L.L. and Tsai W.H.(1991): "Camera Calibration by Vanishing Line for 3-D Computer Vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 4, April, pp 370-6.

[11] Yin J.H., Velastin S.A. and Davies A.C. (1995): "A Calibration Approach for Estimation of Crowd Density Using Image Processing", 2nd International Workshop on Image and Signal Processing: Theory, Methodology, Systems and Applications, 8-10 November, Budapest, Hungary.

Autonomous Road Vehicle Guidance in Normal Traffic

F. Thomanek and E.D. Dickmanns

Institut für Systemdynamik und Flugmechanik
Universität der Bundeswehr Munich
85577 Neubiberg, GERMANY
Tel. +49 89 6004-4147, Fax. +49 89 6004-2082
ld2bms@rz.unibw-muenchen.de

Abstract. A passenger car Mercedes 500 SEL has been equipped with the sense of vision in the framework of the EUREKA-project 'Prometheus III'. Road and object recognition is performed both in a look-ahead and in a look-back region; this allows an internal servo-maintained representation of the entire situation around the vehicle using the 4-D approach to dynamic machine vision. Obstacles are detected and tracked both in the forward and in the backward viewing range up to about 100 meters distance; depending on the computing power available for this purpose up to 4 or 5 objects may be tracked in parallel in each hemisphere. A fixation type viewing direction control with the capability of saccadic shifts of viewing direction for attention focussing has been developed. The overall system comprises about 45 transputers T-222 (16-bit, for edge extraction and communication) and T-805 (32-bit, for number crunching and knowledge processing) and 4 boards based on the Motorola Power Chip (MPC-601) for obstacle detection including image segmentation and state estimation. A description of the parallel processing architecture is given; system integration follows the well proven paradigm of orientation towards 4D physical objects and expectations with prediction error feedback. This allows frequent data driven bottom-up and model driven top-down integration steps for efficient and robust object tracking.
Keywords: Autonomous mobile systems, machine vision, machine perception, data fusion, parallel computing [1]

1 Introduction

Based on experience from equipping two vans, a bus and an all-terrain vehicle with vision systems, in cooperation with Daimler-Benz Research a passenger car Mercedes 500 SEL has been equipped in the framework of the EUREKA-project 'Prometheus III' with a range of sensors for autonomous navigation comprising the sense of vision and inertial sensors for accelerations and angular rates. Eight

[1] This research has been supported by BMFT and Daimler-Benz AG under the EUREKA-Projekt Prometheus III and the EU-Projekt Cleopatra

years of development and testing with the 5-ton van VaMoRs of UniBwM have led to a powerful yet small vision system in the new test vehicle 'VaMP' based on transputers and MPCs; it consists of two sets of cameras fixed relative to each other on a platform for viewing direction control both behind the front and the rear windshield. At least two miniature CCD-cameras will be used on each platform exploiting the bi- or multifocal mode of vision.

Active vision has been expanded to the capability of saccadic vision in order to realize fast attention focussing and high resolution image processing in areas of special interest as, for example, traffic signs and other objects. The 4-D approach allows for intelligent combinations of feedforward and feedback viewing direction control taking egomotion effects into account.

In this paper, a survey is given only; road recognition according to the latest standard is discussed in (Behringer, 1994), obstacle detection, tracking and relative state estimation in (Thomanek et al., 1994), viewing direction control in (Schiehlen & Dickmanns, 1994).

After a brief discussion of the overall system architecture based on spatio-temporal (4-D) processes with orientation towards physical objects and temporal events, the main components will be reviewed: Image acquisition and distribution via the technical eyes and the Transputer Image Processing system (TIP), feature extraction with the software system KRONOS running on 16-bit T-222 transputers as the lowest level of the object processor groups (OPG), the specific OPG's for the object classes 'roads' and 'obstacles', the dynamic data base (DDB) as the means for object-oriented data exchange between the system components, and finally, the component for situation assessment and vehicle control.

2 System architecture

Figure 1 shows the overall system architecture based on the 4-D approach to dynamic machine vision (Dickmanns, 1992); for the first time it encompasses two 'technical eyes' for both forward and rearward perception of the environment. Both eyes will have at least bifocal vision in their final stage of development, very probably even trifocal vision with focal length ratios of three to four; the platforms specifically developed for this application have several modes of operation like inertial stabilization (horizontally), smooth visual pursuit, saccadic shift of attention, and search. Up to four 1/2-inch miniature CCD-cameras may be packed on one platform; the cameras deliver both color and intensity signals which will be used according to the special needs.

The video data are A/D-converted in the framegrabber stage of the TIP (top left); two mono-framegrabbers (MFG) and one color framegrabber are in use at present, each capable of handling up to 4 monochrome and up to 2 color video signals (by time multiplex). These digital data streams are then distributed to the 'Versatile Processing Units' (VPU) taking their needs for object-oriented

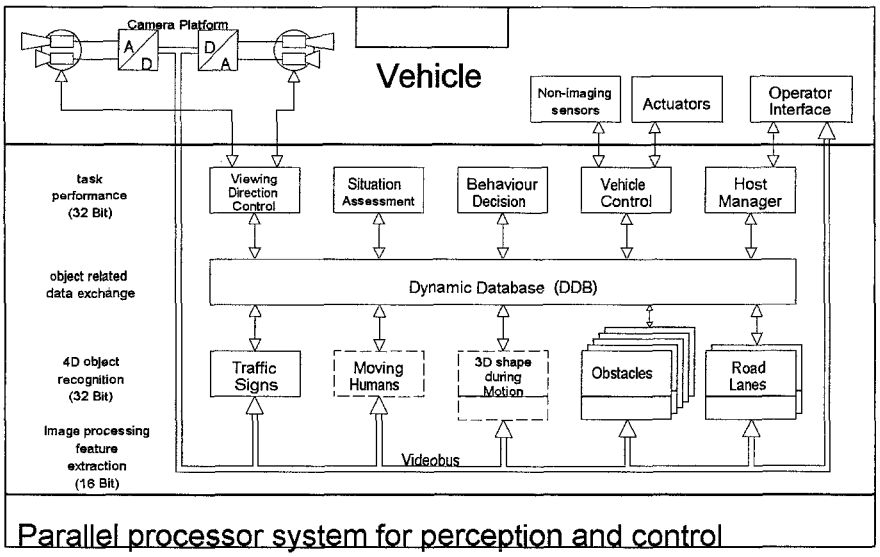


Fig. 1. Overall object-oriented system architecture

data interpretation into account; for example, the 'road detection and tracking'-OPG (RDT) receives data from the wide angle camera for road interpretation nearby and from the tele-camera for more distant road segments (see fig. 2 below). Each RDT-OPG (lower blocks in fig. 1) consists of a VPU for data distribution according to the request of the '4-D object processor' (OP, see below), optionally, the T-222's for image feature extraction with the KRONOS-software, and the upper level 32-bit T-805 object-processors (OP) which implement the recursive state estimation algorithms and the functions of process management for road interpretation (see (Behringer, 1994)).

With the availability of the MPC at the beginning of this year the 'obstacle detection and tracking' (ODT) task moved to this processor. As one MPC is ten times faster than one T805 the edge extraction, object segmentation and state estimation are now performed in 40 ms cycle time for up to 5 targets in one video image.

The results of these OPG's running at 25 Hz processing rate are sent to the DDB (central horizontal bar in fig. 1) for further distribution according to a predefined schedule. The DDB is the exchange platform defined as a standard for implementation-independent data communication; in the Daimler-Benz vehicle VITA_2, also equipped with the core of our transputer system, it is used for coupling other vision subsystems into the overall system; these other sytems may run on different hardware and with different software.

On the higher levels in figure 1 the block 'vehicle control' (VC) is the important component in the autonomous system responsible for generating the correct control outputs; in a more advanced stage it will closely cooperate with the module for 'behavior decision' (BD) basing its output on the results of a module for 'situation assessment' (SA) still under development for VaMP by our Computer Science partners at UniBwM. This architecture has proven successful with the elder VaMoRs- and VITA-systems.

A three-layered approach has been developed for fast and flexible control behavior several years ago at UniBwM (Dickmanns et al., 1994). On the lowest layer, reflex-like behavioral capabilities are implemented based on state vector feedback known to be optimal for linear systems and quadratic goal functions. The 4-D approach to vision yields directly the state variables required for this type of implementation: lane keeping and convoy driving are typical behavioral capabilities realized on this level. This type of implementation yields very fast reaction times and eliminates completely the requirement for action planning for these behaviors. Care has to be taken in order to detect situations in which these simple feedback control laws are no longer applicable. For example, when an obstacle blocks one lane or a forking of the road appears, higher levels capable of understanding the new situation or of coming up with the right decision in the mission context have to override the mode of control running.

In the case of an obstacle in the own lane, and a neighboring lane being free of obstacles, a feedforward generic control time history may be called up with a proper set of parameters which is known to steer the vehicle safely into the neighboring lane; this maneuver may be achieved by applying simple rules to a data set composed of the relative states of several other objects. The upper layer and the middle layer represent this rule based behavior triggered by special events recognized through vision. The transition from free-lane cruising to convoy driving when running up to a slower vehicle in front is another typical event-triggered behavioral component. When a lane change is desirable, it first has to be checked whether this lane change is actually possible without endangering another car; this complex perception task can now be tackled with the capability of rearward viewing. With respect to the task of recognizing objects beside the ego-vehicle in the neighboring lane the two systems being developed jointly with our industrial partner Daimler-Benz (DB) and another university team are different. While DB in their vehicle VITA_2 rely on another two sets of six cameras each on each side of the vehicle for stereo vision (Ulmer, 1994) we prefer saccadic viewing direction control for two sets of cameras on a platform for active vision covering the forward and the rear hemisphere in connection with the 4-D approach in order to accumulate over time the information needed (like humans do by moving their head and eyes). Knowledge about motion processes of bodies with inertia is being exploited systematically in order to make up for the drastically reduced set of data collected.

We will get along with at most 8 cameras in total (actually 4) on two platforms in VaMP while VITA_2 actually has 12 fixed ones just for looking to the sides;

in total, this will lead to at least twice the number of cameras (actually 18), however, with no need for large amplitudes in panning control.

The decision for saccadic vision will affect the overall architecture of the visual perception system deeply; the advantage of a drastically reduced data stream from the vision sensors combined with the need for intelligent control of the perception system over time puts much more emphasis on temporal representations. However, this is not considered to be a disadvantage since intelligence per se needs powerful representations along the time axis which has been overlooked by the AI-approach to vision once before. The engineering sciences have sufficiently powerful tools available for handling the problems encountered.

Our expectation is that the need for temporal smoothing and both inter- and extrapolation over saccadic periods will lead to further developments of the 4-D approach considered mandatory for achieving performance levels approaching the human one in the long run.

3 Actual realisation on transputer-MPC hardware

Figure 2 shows that part of the system architecture realized in VaMP which is directly associated with visual perception; an additional transputer subsystem exists for conventional sensor data processing and actuator control. In total, the system is implemented on a network of about 4 dozen transputers about 2 dozen of which are 16-bit T-222, the rest being 32-bit T-805 and 4 MPC for obstacle detection and tracking.

The Transputer Image Processing system (TIP) may be characterized roughly by the following numbers in our implementation: The video bus data transfer rate of 67 MB/s is used to distribute the images of two cameras connected to each TIP-bus subsystem. At a cycle time of 40 ms, both the tele- and the wide-angle images of size 320 by 256 pixel are transferred to all VPUs connected. Double buffering allows for convenient image access; additional frames for drawing facilitate visualisation of the results on the color graphics displays (CGD, see lower right corner).

With the software package KRONOS each T222 is capable per video cycle of searching about 400 pixel path length in intelligently assigned windows for edges, depending on the mask size; about 20 features usually result which are communicated to the object processor OP directly via links.

3.1 Object Detection and Tracking

The new system, containing 4 MPCs for obstacle detection (instead of 4 T805 + 8 T222), allows a decrease of the cycle time from formerly 80 ms (Thomanek et al., 1994) to 40 ms. An additional object recognition process (Thomanek &

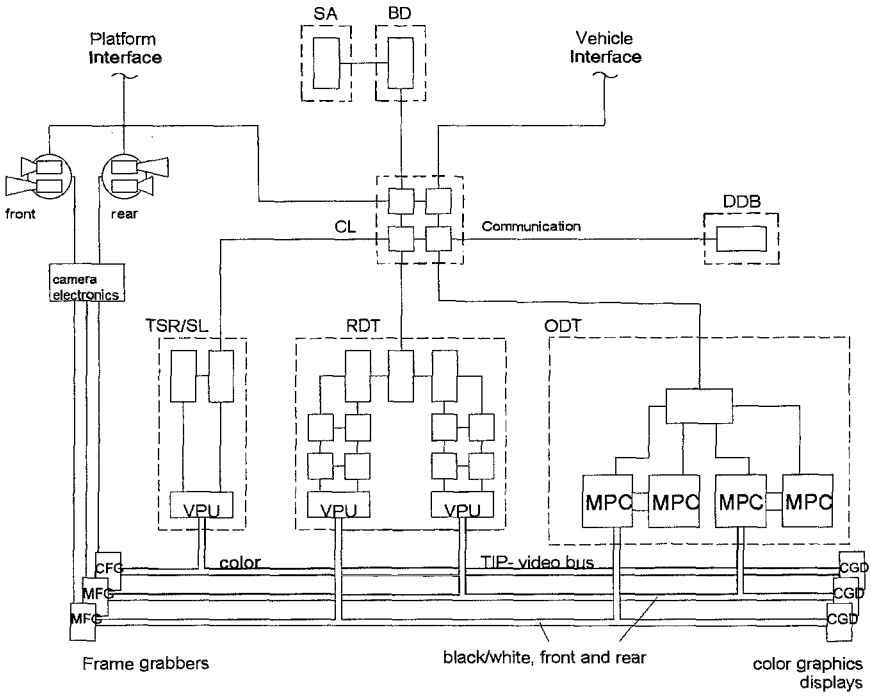


Fig. 2. Transputer/MPC system architecture for visual perception and control

Dickmanns, 1992), running in parallel, leads to high robustness by redundancy as the two segmentation algorithms use different ideas for the detection of obstacles.

In 2 video cycles (40 ms) up to 5 objects may be tracked with one MPC. Object detection in ODT is achieved in 26 vertical search windows covering about 1/3 of the image every 40 ms with gradient masks of size 5 by 3. Up to 180 edges may result which are grouped into up to 20 contours; from these, up to 5 objects are extracted, each characterized by 18 attributes. At 25 Hz evaluation rate this results in 1800 Bytes/second (B/s) data rate per object, i.e. a maximum of 9 KB/s output data rate per OP as compared to the image input data rate of 2.048 MB/s per image, yielding a data rate reduction of 228, hopefully with little loss of information with respect to objects in the scene depicted. More details are given in (Thomanek et al., 1994).

3.2 Road Detection and Tracking

With forward and, for the first time, also rearward recognition of the road, VaMP will be capable of deriving its relative position on the road and in the lane from

interpolation of measurement data; in addition, because of the higher velocities driven, the viewing distance has to be increased. About 200 m viewing range in the direction of the road are desirable and may be achieved with active trifocal vision. In the environmental range then covered (400m) the road may no longer be modelable sufficiently well by just one or two clothoidal arcs moved along the road fixed to the vehicle center.

At high speeds on a straight stretch of road it is very essential to recognize an upcoming curve early and to be able to estimate the true radius of curvature locally correct; under these conditions the simple averaging curvature model of (Dickmanns & Mysliwetz, 1992) is no more sufficient. For this reason, a locally fixed (stationary) piecewise curvature model according to the rules used for construction of high-speed roads is being substituted for the simple (time-varying) sliding model used up to now; this will also simplify the handling of changing numbers of lanes as they occur at entries and exits of highways. This more general model can, lateron, be easily expanded for road forkings and general road crossings.

Another extension that has become necessary with the capability of tracking several objects in each visual hemisphere in parallel and with driving in dense public traffic situations on Autobahnen is to keep track of those image areas where lane and road boundaries are no more visible due to occlusion by these objects. (More than 3000 km have been driven autonomously on public roads since 1992 with the two vans VaMoRs and VITA at speeds up to 80 km/h.)

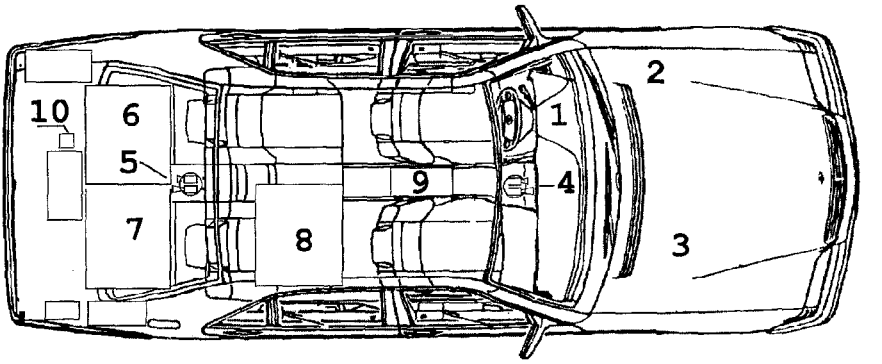
The boundaries of these regions of occlusion may disturb the lane recognition process by simple edge feature machting; therefore, once an object has been detected and recognized, the corresponding predicted image area is blanked out for road recognition.

4 System integration

The installation of the components in the vehicle is shown schematically in figure 3; the trunk of the vehicle is filled with additional electronics in two 19-inch racks (6,7). In order to allow generous equipping of the vehicle with test-, recording- and program development facilities, one of the rear seats has also been taken for installing additional electronic gear (8).

The platforms carrying the cameras are mounted from the top behind the center of the front (4) and rear windshield (5); the inertial sensors are located close to the center of gravity between the two front seats (9). A PC with interface to the transputer/MPC system serves as host systems and man/machine interface.

System integration has been performed in summer 1994 in parallel to the companion system VITA_2 of Daimler-Benz. Both vehicles have been official Common European Demonstrator vehicles 'Obstacle Avoidance' (CED-3) for the final



1 electrical steering motor
2 electrical brake
3 electronic throttle
4 front pointing platform for CCD cameras
5 rear pointing platform
6 Transputer/MPC Image Processing system
7 platform and vehicle controllers
8 electronics rack, human interface
9 acceleratometers (3 orthogonal)
10 inertial rate sensors

Fig. 3. Top view of VaMP, components for autonomous driving

Demo of the EUREKA-project 'Prometheus' in October 1994. In summer 1995 the portation of ODT on MPC-boards was finished; road recognition will follow soon.

5 Conclusions

With VaMP of UniBwM and the twin vehicle VITA_2 of Daimler-Benz test vehicles of advanced performance levels with slightly different perception systems are available which allow to further explore the promising new technology of dynamic machine vision for autonomous road vehicle guidance. With the MPC processors the cycle time of the ODT module decreases from 80 ms to 40 ms und the amount of space from 4 to 2 boards.

The new generation of microprocessors allow a further increase of system capabilities over the next years while simultaneously shrinking both volume and power consumption of the system. Both the perception subsystems and the overall system architecture are becoming more and more stable and powerful, which is a good indication of the systems becoming more mature. However, it should not be overlooked that in order to achieve robust performance approaching the performance level expected from human drivers there is still a long way to go.

References

Behringer, R. (1994). Road Recognition from Multifocal Vision. In: *Proc. of the Int. Symp. on Intelligent Vehicles.* Paris,France. pages 302–307.

Dickmanns, E. D. (1992). Machine perception exploiting high-level spatio-temporal models. In: *AGARD-Lecture Series 185.* Hampton USA; Munich - Madrid.

Dickmanns, E.D. & Mysliwetz, B. (1992). Recursive 3-D state and relative ego-state recognition. *Pattern Analysis and Machine Intelligence* 14(2), 199–213.

Dickmanns, E.D., Behringer, R., Dickmanns, D., Hildebrandt, T., Maurer, M., Schiehlen, J. & Thomanek, F. (1994). The Seeing Passenger Car VaMoRs-P. In: *Proc. of the Int. Symp. on Intelligent Vehicles.* Paris,France. pages 68–73.

Schiehlen, J. & Dickmanns, E.D. (1994). A Camera Platform for Intelligent Vehicles. In: *Proc. of the Int. Symp. on Intelligent Vehicles.* Paris,France. pages 393–398.

Thomanek, F. & Dickmanns, D. (1992). Obstacle Detection, Tracking and State Estimation for Autonomous Road Vehicle Guidance. In: *Proc. of the 1992 Int. Conf. on Intelligent Robots and Systems.* IEEE, SAE. Raleigh NC. pages 1399–1407.

Thomanek, F., Dickmanns, E.D. & Dickmanns, D. (1994). Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance. In: *Proc. of the Int. Symp. on Intelligent Vehicles.* Paris, France. pages 231–236.

Ulmer, B. (1994). VITA II - Active Collision Avoidance in Real Traffic. In: *Proc. of the Int. Symp. on Intelligent Vehicles.* Paris,France. pages 1–6.

Evaluation of Image Sequences from Outdoor Scenes: Selected Problems and Solutions

Hans-Hellmut Nagel [1,2] *and Henner Wolfgang Kollnig* [1]

[1] Institut für Algorithmen und Kognitive Systeme
Fakultät für Informatik der Universität Karlsruhe (TH)
Postfach 6980, D-76128 Karlsruhe / Germany
and
[2] Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB)
Fraunhoferstr. 1, D-76131 Karlsruhe / Germany (e-mail: hhn@iitb.fhg.de)

Abstract. Lane structures are estimated, based on image sequences recording complex inner-city intersections, by a model-based approach. First, by estimating vanishing point coordinates from straight line segments extracted from the image sequence, an initial guess regarding the orientation of the camera with respect to the road plane as well as regarding the focal length is obtained. Subsequently, various steps of both model and process refinement result in a significantly improved lane structure estimation. Results from experiments with real world inner-city multi-lane intersection scenes are presented.

1 Introduction

Several research groups, although for different purposes, recently have taken up the challenge to determine the lane structure of an intersection by evaluation of monocular image sequences. A primary application relates to automatic, vision-based driving. On the one hand, an intersection has to be detected whenever an automatically driving vehicle approaching the intersection does not have the right of way: the intersection can only be crossed provided no other vehicle has priority. In order to detect vehicles on intersecting roads, it is an advantage to localize these roads first, i.e. the intersection lane structure has to be extracted from the image sequence which provides the input for the lateral guidance during uninterrupted stretches of the road to be followed - see, e.g., [3]. On the other hand, the intersection structure must be known if an automatic vehicle has to turn from the road on which it approaches an intersection, onto a joining road. Even if such a maneuver can be based on the support of a map-guided navigation system, the turning maneuver itself can be performed much more reliably if images can be taken and evaluated continuously for the particular course to be followed, especially if one or both lane marking(s) along the desired trajectory leave the field of view on very sharp corners ([9, 10]).

A quite different problem is defined by the task to detect and track vehicles along complex road intersections, for example in order to determine finely differentiated traffic behavior. Vehicle tracking at intersections based on computer

Fig. 1. First frame of image sequence 'F' recorded at an inner-city intersection.

vision has been studied, for example, by [4, 5, 7]. This work relies on a map of the intersection and its markings so that the video camera could be calibrated by interactively associating feature points in the map to their image points in a video frame. If new image sequences of road traffic at inner-city road intersections have to be evaluated, the unavailability of the corresponding intersection maps will jeopardize the application of their approach.

The investigations to be reported here discuss a solution out of this predicament. We use a coarse 3D lane structure model in order to automatically obtain even the initial estimate for the camera position and orientation. For this purpose, edge elements extracted from an image are clustered into line segments. Subsets from these line segments are extracted which hypothetically belong to the images of parallel lines in the scene. The vanishing point for a subset of images of hypothetically parallel lines is then determined. Most of the camera calibration data – including its position and orientation in the scene – can be estimated once we have succeeded in obtaining the image coordinates of at least two vanishing points corresponding to two different orientations in the plane of the road intersection.

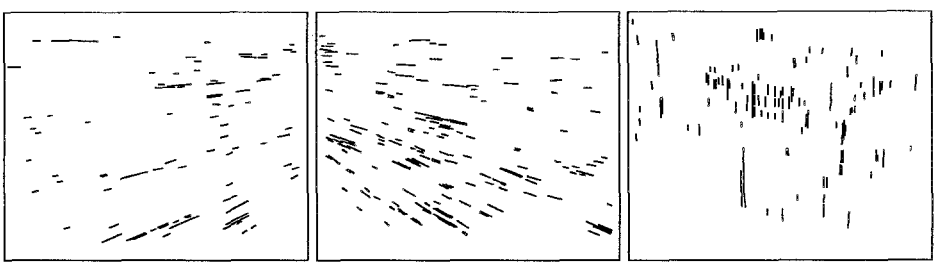
Since the estimation of the camera parameters has not been an end in itself, we naturally reevaluated the estimation process sketched above whenever we came across unsatisfactory tracking results obtained with these camera parameters. As a consequence, the – only coarsely outlined – estimation process underwent a number of developmental refinements until it reached the status to

be reported here. The various steps of this refinement process are reflected in the titles of subsequent sections and subsections, illustrated by a new recorded test image sequence of a much frequented multi-lane inner-city street intersection (see Figure 1).

2 Fit of the Intersection Model to an Image Sequence

2.1 Obtaining Initial Guesses for the Camera Parameters

The perspective projections of parallel scene lines into the image plane intersect in a *vanishing point*. The vector from the optical center towards the vanishing point is parallel to the 3D orientation of the parallel scene lines corresponding to this vanishing point. In our approach, we first determine two rotational camera parameters using the azimuth angles of the three principal vanishing point directions, then the focal length [6]. After having computed the orientation and the focal length, the image line segments are regrouped similarly to the technique proposed by [8]. In contrast to [8], however, we do not exploit this technique for grouping edge elements, but only in order to *iteratively* improve our calibration parameters. Moreover, in contrast to [8], we do not have estimates either of the camera orientation from other sensors, e.g. odometer or inclinometer, or of the focal length. The result of the regrouped line segments can be found in Figure 2. Finally, the scale parameter is determined.



(X) (Y) (Z)

Fig. 2. After having computed initial guesses for the orientation and the focal length of the camera, the image line segments are regrouped according to the principal vanishing points of the X, Y and Z axis of the world coordinate system, respectively.

In our investigation it turned out that the initial guesses about the orientation of the road plane with respect to the camera are not exact enough in order to restrict the pose estimation process to the estimation of three degrees of freedom, using a ground plane constraint as proposed by, e.g., [2]. It is thus necessary to admit all six pose parameters for adaptation in the iterative fit versus only three parameters in an initial version. The three rotational and three translational

parameters to be estimated are combined in a six dimensional state vector. It is estimated using a recursive maximum a posteriori (MAP) estimator similar to an approach proposed by, e.g., [4] exploiting straight line segments extracted from the current frame as measurements.

2.2 Deletion of Edge Segment Center and Length from the Fitting Process

In many cases, the good results obtained by, e.g., [4] with the edge segment representation of [1] could not be repeated in the current application while fitting an intersection model to frames of a sequence recording a road intersection, since sections and, therefore, border and separation lines are much longer than the lines of a vehicle image at the same scale. There is, moreover, a high probability that the image of a road boundary or a separation line is partially occluded, for example by vehicles, lamp posts, or flagpoles. As a consequence, the probability is much lower that a data segment can be extracted corresponding to the full length of a road marking or curb line. The midpoint position and the length of extracted data segments thus often show only a small correlation with the projected center position and length of the corresponding model segment.

A different problem causes the same trouble for the quantitative evaluation of a match between data and model segments. In the case of interrupted markings between neighboring lanes, the end points and the length of the corresponding data segments could often be extracted reliably from an image. But in this case it is the exact position of the corresponding model segment which is unknown. As a consequence, it appeared more appropriate to no longer exploit the center position and length of a data segment during the fit. The Mahalanobis distance between a model and a data segment thus has been reduced to only two contributions, namely the difference in orientation between a data and a projected model segment in the image plane, and the normal distance between the center point of a data segment and the (tentatively) matched model segment. This at least avoided a bias due to errors in the localization of end points of data segments.

2.3 Taking the Finite Width of Road Markings into Account

Another effect could be identified which occasionally caused trouble, namely extracting parallel edge segments which correspond to both gray value transitions of a road marking, the one from dark to bright gray values as well as the bright to dark transition at the other side of the marking. Only a single data segment, corresponding to either one of these transitions, had been used initially. The restriction to a 1:1 association between model and data segments is justified for vehicles, at least in the case of the image sequences evaluated so far. It is not, however, justified in the case of road markings if their images appear broad enough to facilitate the extraction of separate edge segments for either side of a marking. Admitting many-to-one associations between data segments and a model segment would appear to solve this problem. It did not: since the data

driven edge segment extraction process can not guarantee to deliver at most two data segments for each model segment and, moreover, in a more or less symmetric position with respect to the desired model segment, admission of many-to-one matching grossly deteriorated the performance of the iterative fitting process. This simple approach had to be abandoned quickly.

2.4 Taking Contrast Direction Across Edge Segments into Account

Rather than returning to the original 1:1 matching between data and model segments, it appeared more promising to more fully exploit the knowledge about the structure of road markings in the scene: it is known to represent a band of a certain width, being brighter than the road background. The model of a marking for the side of a lane, therefore, is no longer treated as a mathematical line in the scene corresponding to the center of the lane marking. It is now modeled as a bright band with a width which can be read from the book of regulations for road markings, about (0.12 ± 0.02) m. This refined model facilitated to formulate where in the image relative to the projected center line of the lane marking one should expect an edge segment corresponding to a dark-to-bright transition and where one should expect to find an edge segment corresponding to the opposite transition from the bright interior of the marking band to the darker road surface. It turned out that just this requirement alone reduced the combinatorial explosion sufficiently while matching data segments to model segments. It even facilitated the admission of *many-to-one matching*. Since in this manner significantly more data segments could be taken into account, the fit of the intersection model to an image of the intersection has been much improved.

2.5 Multi-Frame Fit of the Intersection Model to an Image Sequence

So far, only the fit of the intersection model to a single image frame from a sequence has been treated. Closer inspection showed that the result can be influenced by the inclusion of matches between model segments and some data segments extracted from a region corresponding to the image of a vehicle – see Figure 3.

In principle, one can not exclude the possibility that an uninformed matching process attempts to associate model segments to data segments which do not belong to a road marking or to curbs of a road. There is, however, a fairly general and nevertheless powerful heuristic which facilitates coping with this kind of problem.

The basic observation consists in noting that the image position of **moving** vehicles will shift with the frame number whereas, in the case of a stationary video camera, the image of edge segments belonging to the intersection markings should remain at the same scene – and hence image – position. We, therefore, just extended a tracking approach, for example the one described by [4], to the intersection model, postulating constant motion with zero magnitude as a motion model for the intersection relative to the camera. Unless an initial fit is severely

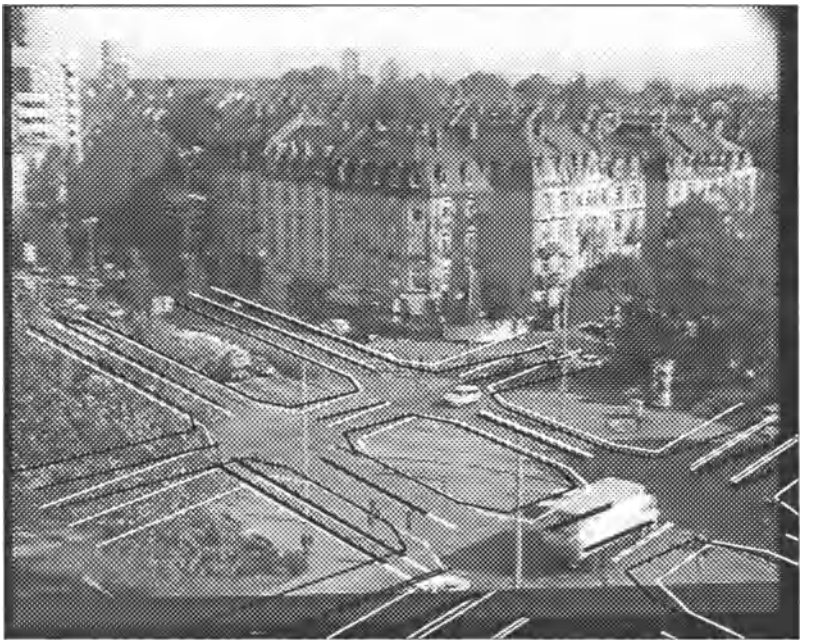


Fig. 3. Fit of an intersection model to a single image from a sequence. The mismatch between model segments and the strong edge data segments from the van in the lower right of the image frame resulted in a distorted position estimate for the intersection model. The projection of the initial intersection position into the image plane is indicated by white lines, the projection of the model position after the fit by dark lines.

distorted by a mismatch between data and model segments, one can expect that the results obtained after all iterations within one frame represent at least an usable estimate. If this state estimate is subsequently used as the starting point for the prediction of the intersection position at the next frame time by a Kalman Filter, the covariance matrix for the camera parameters will be updated after exploitation of observations at the next frame-time. Edge segments associated with moving vehicles will only have a comparatively small effect after a few frames since there is constant reinforcement in the direction of uncorrupted values by all matches between model segments and data segments which are correctly associated with road markings and borderlines. As a consequence, the elements of the covariance matrix for the camera parameters will tend to become smaller and thus the influence of temporally varying mismatches between model segments and data segments originating from the images of moving vehicles will diminish: no new mechanisms need to be introduced, a simple variant of the approach which has been developed for the tracking of moving vehicles is sufficient (see Figure 4). The confidence we had gained in our ability to detect and properly interpret systematic differences between the intersection model

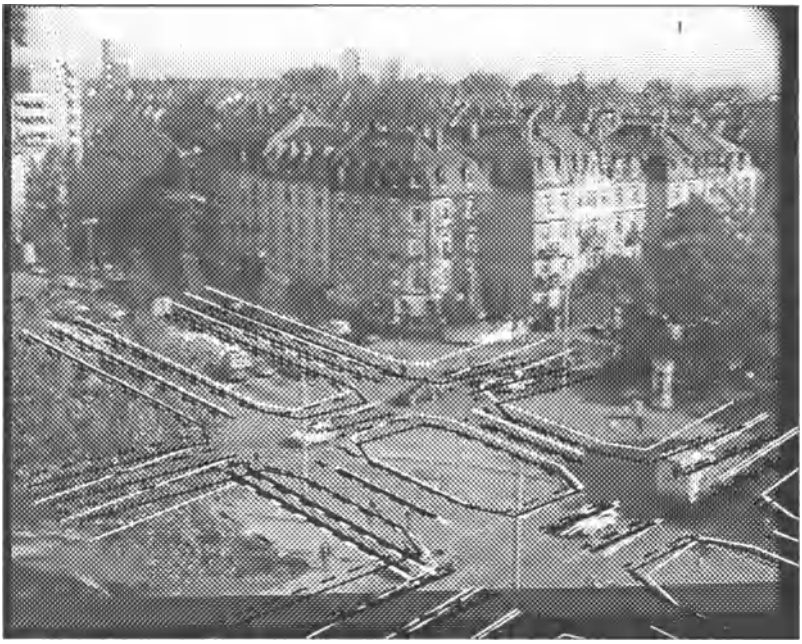


Fig. 4. Result obtained after predicting and correcting the intersection pose in the scene on the basis of zero motion relative to the recording camera through 89 halfframes. The projection of the initial intersection position into the image plane is indicated by white lines, the projection of the model position after the fit by dark solid lines. Moreover, the projection of the model position after the evaluation of this single image *without* the knowledge obtained by a Kalman Filter based prediction step is indicated by dark dashed lines. It can be seen that the initial distortions due to mismatches between segments of the intersection model and data segments originating from moving vehicles have diminished considerably.

and its fit to the image sequence let us eventually suspect that the two road systems do not intersect each other at exactly a right angle. It turned out that we eventually succeeded to obtain an official map of this intersection and could indeed verify that the smaller angle between the two road systems is closer to eighty degrees.

3 Exploiting the Lane Structure for the Initialization of Vehicle Tracking

The tracking approach described by [4] depends on fitting the projection of a 3D vehicle model into the image plane to edge segments extracted from the current image frame. This nonlinear process requires a proper initialization in order

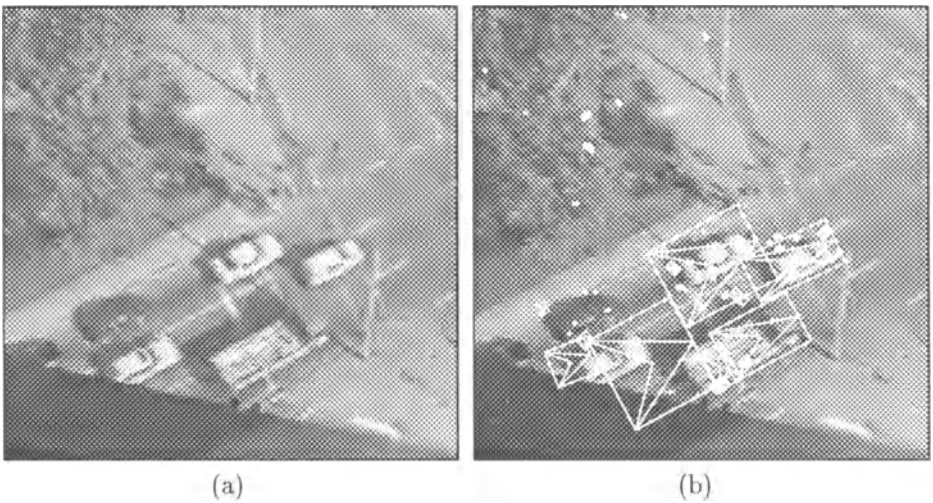


Fig. 5. (a) Section from the lower left corner of frame 220 of another sequence from the same intersection illustrated in Figure 1. (b) Hypotheses for image regions corresponding to a moving vehicle, superimposed to the image section shown in (a).

to offer a good chance to converge towards the desired optimal fit. As long as vehicle images are sufficiently well separated from each other, the initialization procedure provided good starting values for the unknown parameters, namely position, orientation, and velocity of the vehicle model on the road plane in the 3D scene. In case several parallel queues of vehicles waiting in front of a traffic light begin to move once the traffic light switches to green, one frequently encounters a situation where two neighboring vehicles move in the same direction with approximately the same speed. In such a situation, the optical flow field segmentation process will return a single cluster for the combination of two neighboring vehicles – with the consequence that the initialization step is grossly misled and the tracking process fails sooner or later, Figure 5.

Tightening the tolerances which control the selection and clustering of neighboring optical flow vectors does not present a viable solution: the optical flow field for longer vehicles with more or less homogeneously colored surface facets may break up into two or more clusters which again present difficulties for a proper initialization. The problem is not the inappropriate choice of parameter values, but an algorithm built on insufficient knowledge about the situation in the scene.

After having determined the lane structure of the intersection as described above, it becomes possible to decompose a cluster of optical flow vectors into two parts each of which is assumed to correspond to the projection of a vehicle onto a single lane. Since it is possible to estimate the general direction of vehicular traffic on a lane, one may even tighten the hypothesis formulation:

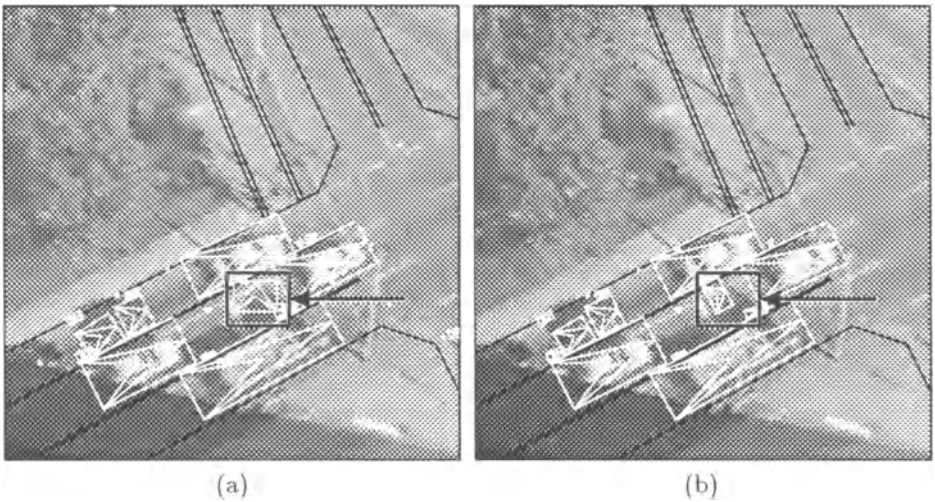


Fig. 6. (a) Initial hypotheses for vehicle images obtained from optical flow vectors estimated from a subsequence around the image section shown in the lower left corner of frame 220 of the sequence illustrated in Figure 5; it can be seen that the hypothesis enclosed by a dark rectangle exhibits a motion direction – indicated by the white arrow inside the white rectangle representing this hypothesis – which is incompatible with the traffic direction for this lane. The optical flow vectors contributing to this hypothesis are associated with gray value transitions related to the shadow cast by the truck in the lower lane. (b) Hypotheses for moving vehicles retained under the provision that only velocity estimates compatible with the traffic direction for the corresponding lane are admitted.

only those initializations are retained which exhibit compatibility between the average direction of the optical flow vectors contributing to a cluster and the traffic direction of the lane on which a model position is initialized. Figure 6 shows the improvement which can be obtained by the requirement that a vehicle model must not cover two lanes and by the additional restrictions resulting from a compatibility requirement between an initial estimate for the vehicle velocity and the general direction of traffic in the lane.

An exemplary result of model-based tracking, exploiting the approach by [7] and based on the solutions outlined above, is illustrated in Figure 7.

Acknowledgment

We thank Katharina Ehmann, Michael Haag, Holger Leuck, and Till-Henning Osswald for their support and stimulating discussions about the topic of this contribution.

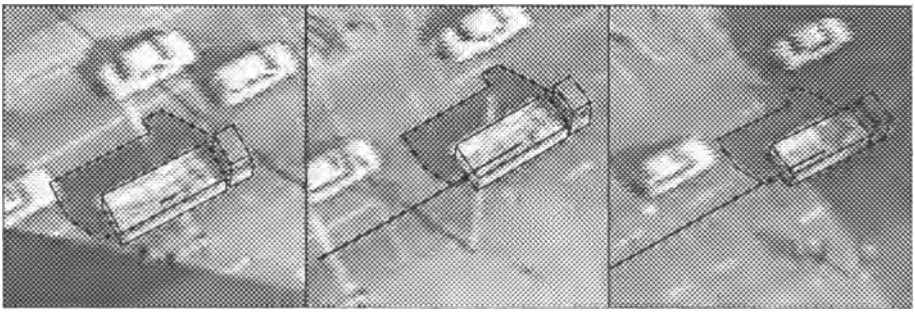


Fig. 7. The evaluation of the image sequence depicted by its 213rd, 305th, and 433th frame. The shadow contour is used in the pose estimation process.

References

1. R. Deriche, O. Faugeras: *Tracking Line Segments.* Image and Vision Computing **8**:4 (1990) 261–270.
2. L. Du, G.D. Sullivan, and K.D. Baker: *Quantitative Analysis of the Viewpoint Consistency Constraint in Model-Based Vision.* Proc. Fourth International Conference on Computer Vision (ICCV '93), 11-14 May 1993, Berlin/Germany, pp. 632–639.
3. K. Kluge and C.E. Thorpe: *Intersection Detection in the YARF Road Following System.* Intelligent Autonomous Systems IAS–3, 15–18 February 1993, Pittsburgh / PA, F.C.A. Groen, S. Hirose, C.E. Thorpe (eds.), IOS Press Washington / DC 1993, 145–154
4. D. Koller, K. Daniilidis, H.-H. Nagel: *Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes.* International Journal of Computer Vision **10**:3 (1993) 257–281.
5. H. Kollnig, H.-H. Nagel, and M. Otte: *Association of Motion Verbs with Vehicle Movements Extracted from Dense Optical Flow Fields.* ECCV '94, Vol. Two, J.-O. Eklundh (ed.), Lecture Notes in Computer Science **801**, Springer-Verlag New York/NY 1994, 338–347
6. H. Kollnig: *Ermittlung von Verkehrsgeschehen durch Bildfolgenauswertung.* Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe / Germany, February 1995. See: *Dissertationen zur Künstlichen Intelligenz* DISKI **88**, infix, Sankt Augustin / Germany 1995 (in German)
7. H. Kollnig, H.-H. Nagel: *3D Pose Estimation by Fitting Image Gradients Directly to Polyhedral Models.* Proc. Fifth International Conference on Computer Vision (ICCV '95), Cambridge/MA, 20-23 June 1995, pp. 569–574
8. X. Lebègue, J.K. Aggarwal: *Significant Line Segments for an Indoor Mobile Robot.* Trans. Robotics and Automation **9**:6 (1993) 801–815.
9. G. Struck, J. Geisler, F. Laubenstein, H.-H. Nagel, and G. Siegle: *Interaction between Digital Road Map Systems and Trinocular Autonomous Driving.* Intelligent Vehicles '93 Symposium, July 1993, Tokyo/Japan, pp. 461–466
10. G. Struck, J. Geisler, F. Laubenstein, H.-H. Nagel, and G. Siegle: *Multi-Camera Vision-Based Autonomous Maneuvering at Road Intersections.* Intelligent Vehicles '94 Symposium, October 1994, Paris/France, pp. 189–194

Vehicle Localisation and Discrimination in Outdoor Traffic Scenes

T N Tan, G D Sullivan and K D Baker

Department of Computer Science, University of Reading,
Reading, Berkshire RG6 2AY, England

Abstract. This paper concerns model-based vision for road traffic scene analysis. A traffic vision system has recently been developed at The University of Reading. The three main modules of the system are Movement Detection, Vehicle Localisation and Discrimination, and Vehicle Tracking. This paper outlines our work on the localisation and discrimination module. Emphasis is on recovering 3D poses of road vehicles in given image regions. Two classes of algorithms are described, one based on symbolic image features (line segments), and the other simply on image intensity gradients. *A priori* knowledge about traffic scenes and vehicles is exploited to improve the performance and efficiency of the algorithms. The algorithms are tested extensively with routine outdoor traffic images, and examples are included to demonstrate their principles.

1 Introduction

Traffic is an important issue in a modern society, and the automatic monitoring and surveillance of road traffic is of increasing commercial importance. Machine vision offers unrivalled opportunities for wide-area assessment of traffic. Such opportunities have generated a growing interest in visual traffic scene analysis in the computer vision community [1-6]. The work presented in this paper is the result of several research projects conducted at the University of Reading, the main goal of which has been to investigate the hardware/software implementation of a model-based traffic vision system [2]. The current state of implementation is a software realisation of the system capable of tracking a single vehicle in real-time [7]. Fig.1 is a schematic diagram of the system showing dataflow between various modules.

The three major modules of the system are Movement Detection, Vehicle Localisation and Discrimination, and Vehicle Tracking. Camera models (i.e., intrinsic and extrinsic camera parameters) and vehicle models (i.e., 3D geometric descriptions of vehicles) are established off-line. The Movement Detection module takes input from a CCD camera and identifies image regions (called regions of interest - ROI) likely to contain road vehicles. The next module, Vehicle Localisation and Discrimination, then processes data in each ROI so as to determine the 3D pose and class of the detected vehicle. The output of the Localisation and Discrimination module bootstraps the Tracking module which then tracks each vehicle automatically based on the pose and class information in the previous frame and known vehicle dynamics.

In the subsequent discussions we focus our attention on the Localisation and Discrimination module. We outline a number of algorithms for vehicle localisation. Performance with routine outdoor traffic images and computational efficiency are

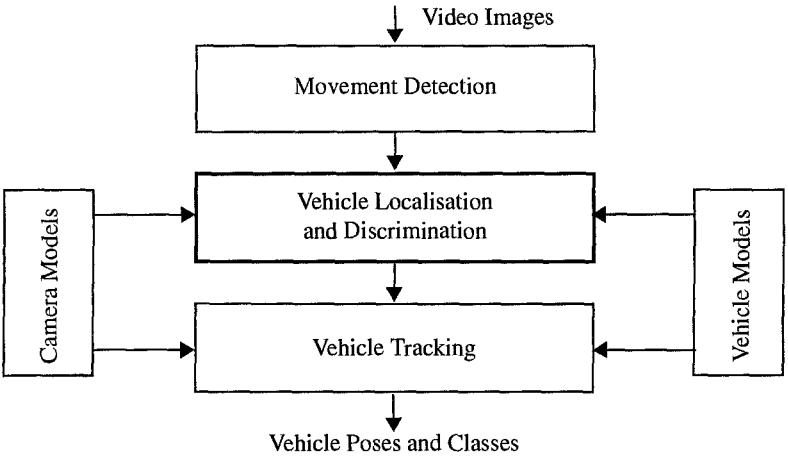


Fig. 1. Schematic diagram of the model-based traffic vision system.

the two major factors we have considered while designing the algorithms. Due to space limitation, details of these algorithms cannot be provided here but may be found in the appropriate references. Likewise, readers interested in other aspects of the model-based traffic vision system are referred to elsewhere [2, 7].

2 Geometric constraints in traffic scenes

There are several physical and geometric constraints in a typical traffic scene which can be exploited to improve the robustness and efficiency of vision algorithms.

2.1 Ground-plane constraint

The current implementation of the vision system uses a single static calibrated camera. Within the field of view of the camera, the ground surface is approximately planar. Furthermore, under normal conditions vehicles have their wheels in contact with the ground-plane (GP). This fact, which we call the ground-plane constraint (GPC) [8, 14], reduces the degrees of freedom (DoF) of a vehicle from 6 to 3 (we assume that the vehicle body is rigid and changes in height due to suspension are negligible). By choosing appropriate vehicle model coordinate system (MCS) and world coordinate system (WCS), the 3 DoFs (i.e., the 3D pose) of the vehicle can most conveniently be represented as the location (X, Y) on the GP and the orientation (θ) w.r.t. the normal of the GP [9-10].

2.2 The shoebox model

The appearance of most vehicles is dominated by two sets of parallel lines - one set along the width direction (left-right) and the other along the length direction (front-rear). The MCS is defined so that the widthwise set is aligned with the X-axis and the lengthwise set with the Y-axis of the MCS. We denote the two sets by M_x and M_y respectively. Without considering size and relative positions, M_x and M_y form

a generic model for road vehicles - the shoebox model. This model, as shown later in this paper and in [12], allows the orientation of vehicles to be determined very efficiently and independently of specific models.

2.3 The weak perspective assumption

In a typical camera set-up in outdoor traffic scenes, the distance from a vehicle to the camera is usually much greater than the diameter (the maximum dimension) of the vehicle. This allows the perspective projection to be approximated well by the weak perspective projection. The weak perspective assumption (WPA) further reduces the computational cost of the vision algorithms [11].

In the following sections, we show how the above physical and geometric constraints in traffic scenes are used to simplify vehicle localisation and discrimination. Two groups of algorithms are described, one based on image line segments and the other on image intensity gradients.

3 Line-based vehicle localisation

We first discuss the line-based algorithms. It is well-known (e.g., [13, 16]) that a given match between a 2D image line and a 3D model line (called a 2D-3D line match) provides two independent constraints on the object pose. Under the GPC, the two constraints are particularly simple and given by

$$\begin{cases} F\cos\theta + G\sin\theta = H \\ A\cos\theta + B\sin\theta + CX + DY = E \end{cases} \tag{1}$$

where A-H are terms computable from known variables such as the image and model coordinates of the image and model lines (see [9-10, 17]). The first constraint only involves the orientation parameter θ. This allows vehicle orientation (θ) and vehicle location (X, Y) to be determined separately in two successive steps.

3.1 Orientation recovery

For each 2D-3D line match, the orientation constraint can easily be solved to obtain two orientation values. When data is noisy, the two values are replaced by two orientation intervals where the probability density functions (PDF) of the orientation parameter can be derived from known noise models [9-10].

To determine the orientation of a known object/model in a given ROI, 2D image line segments are first extracted and then matched with the line segments of the 3D model. Since we do not know which image line corresponds to which model line, each 3D model line is considered as a match for each image line. For each hypothesized match, the PDFs of the orientation parameter are computed. Solutions from all possible 2D-3D line matches are then accumulated. This gives a 1D distribution (called the orientation histogram) of the evidence in favour of the parameter taking a particular value. The peaks of the histogram identify the most likely orientations of the vehicle. An example illustrating the above ideas is shown in Fig.2, where the four major peaks of the orientation histogram are located

approximately 90° apart from each other due to the rectilinear nature of the vehicle model.

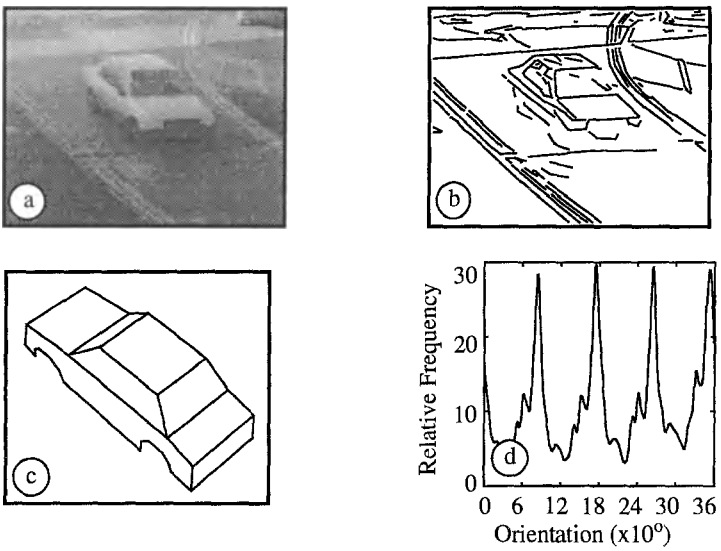


Fig. 2. Recovery of object orientation. (a) ROI (350x270 pixels); (b) extracted line segments; (c) a saloon vehicle model; and (d) orientation histogram.

The orientation recovery process outlined above not only determines the orientation it also identifies a sub-set of line matches consistent with the peak orientation.

3.2 Determination of GP location

Once the orientation θ is known, the vehicle is confined to slide along a confusion line on the GP defined by the second constraint of (1). The possible positions of the object on the GP can be further restricted by requiring the projection of the model line to overlap the corresponding image line segment. The confusion line is therefore bounded [9-10].

Then for each orientation peak, we discard all line matches except those which contribute to it. Each of these matches constrains the location of the origin of the MCS to lie on a bounded confusion line. In this way, each accepted line match defines a segment of line in the remaining configuration space (the GP, (X,Y)). All confusion line segments are tallied in an X-Y accumulator array, and subsequently blurred by a Gaussian. Fig.3(a) shows the X-Y accumulator array so obtained for the global orientation peak ($=175^o$) of Fig.2(d). Darker points in Fig.3(a) indicate higher accumulator values. Local maxima in the accumulator array are identified to recover the most likely locations of the vehicle (having the known orientation). In the case of an isolated vehicle such as Fig.2(a), we usually obtain one conspicuous peak as seen in Fig.3(a). Fig.3(b) shows the saloon model instantiated in the image in this pose (i.e., the global peak pose). The fit is very close.

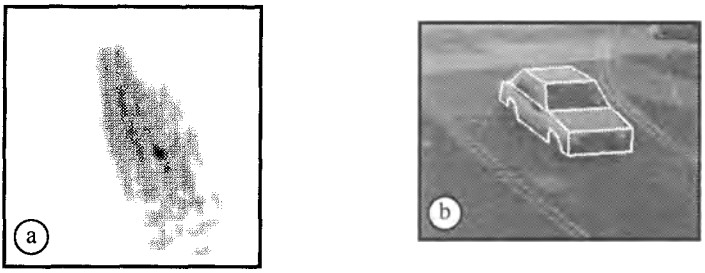


Fig. 3. Recovery of GP location. (a) X-Y accumulator array at global peak (=175°) of Fig.2(d); (b) pose corresponding to the global peak of the array.

4 Gradient-based vehicle localisation

The algorithms outlined in the preceding section entail image line segment extraction. The process of line extraction is typically error-prone and time-consuming. Its computational complexity is also data-dependent. This makes the real-time implementation of the algorithms difficult. In this section we introduce the second group of vehicle localisation algorithms which do not require symbolic image feature extraction. The algorithms only need access to local gradient data which can easily be obtained using a gradient operator (e.g., the Sobel operator).

4.1 Orientation recovery

With known horizontal and vertical gradient d_u and d_v at each pixel (u, v) in the ROI, we can construct a virtual image line that passes through the pixel and is perpendicular to the gradient direction. We then match all virtual image lines with the 3D model lines and follow the procedure of Section 3.1 to obtain the most likely orientations.

The recovery process is made model-independent by adopting the shoebox model. In particular, if d_u and d_v at pixel (u, v) are due to the projection of a model line belonging to one of the two parallel sets, then the two possible orientations for the object are either θ and $180^o + \theta$, or $90^o + \theta$ and $270^o + \theta$, where θ is computed as follows

$$\theta = arc\tan\left(\left|\frac{(m_{11} - um_{14})\, d_u + (m_{13} - vm_{14})\, d_v}{(m_{21} - um_{24})\, d_u + (m_{23} - vm_{24})\, d_v}\right|\right) \tag{2}$$

where m_{ij} is the element at the ith row and the jth column of the known perspective transformation matrix [12].

By considering the gradient vector at each pixel as the possible consequence of the projection of M_x and M_y, an algorithm similar to that described in Section 3.1 can be devised. The algorithm is now, however, model independent, and is therefore significantly more efficient. An example is shown in Fig.4, where the periodicity of the orientation histogram is due to the 90°-rotation symmetry of the shoebox model.

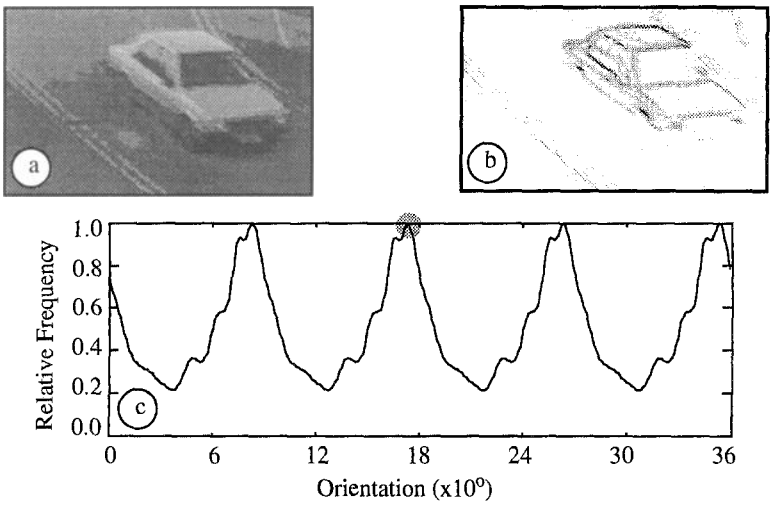


Fig. 4. Gradient-based model independent orientation recovery. (a) ROI (322x195 pixels); (b) Sobel gradient image; (c) orientation histogram.

4.2 Determination of GP location

With WPA and a given orientation, the difference in the image-plane projection of an object model caused by the GP location (X, Y) will generally be small unless the image region is excessively large. We assume that the difference is negligible since we are only concerned with relatively small regions of interest (ROI). A representative projection can therefore be obtained by instantiating the object model at the known orientation and the centre of the ROI.

The 2D model projection is regarded as a gradient image and is collapsed along a number of directions to obtain a set of 1D model profiles. Similar projection operations are applied to the intensity gradient image of the ROI to get a set of 1D data profiles. The peaks of the cross-correlation functions of the corresponding model and data profiles specify a set of confusion lines on the image plane on which the projection of the origin of the MCS might lie [11]. The intersections of the confusion lines then identify the likely GP locations of the object. An example is given in Fig.5 and Fig.6, where the model profiles shown in the middle row of Fig.5 correspond to the saloon model (see Fig.2(c)) instantiated at the centre of Fig.4(a) and the orientation marked in Fig.4(c). It can be seen that the three confusion lines are very consistent, and the recovered pose is very accurate.

4.3 Remarks

The computational cost of orientation recovery can be trivialized by combining the shoebox model and the WPA. It is shown in our previous work [11] that under the WPA, the orientation can be determined merely by matching the major gradient directions of the ROI with the directions of the model lines. This typically means matching 5 gradient directions with the 2 shoebox directions.

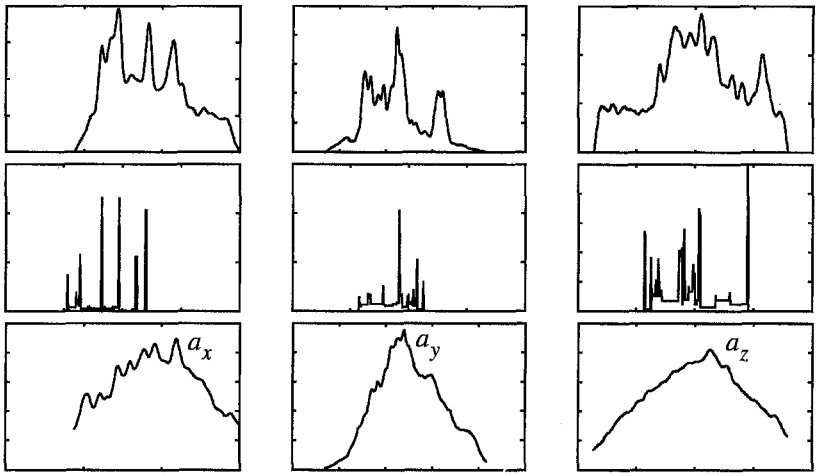


Fig. 5. Model and data profiles and their correlations for ROI shown in Fig.4. Top row: data profiles; middle row: model profiles; bottom row: correlation functions between model and data profiles.

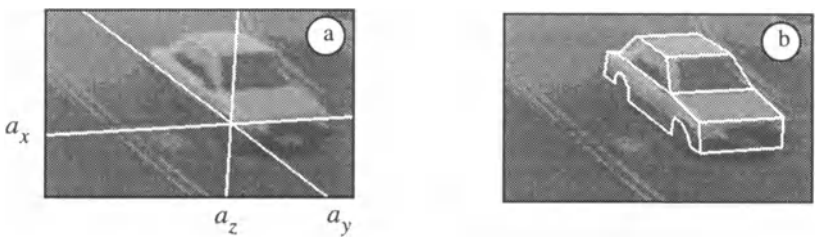


Fig. 6. Recovery of GP location via 1D correlation. (a) three confusion lines; (b) a saloon model at the recovered pose.

5 Pose evaluation and discrimination

For an object in a given image region, the algorithms described in the last two sections will in general generate multiple candidate poses (e.g., due to multiple peaks of the orientation histograms, the X-Y accumulator arrays, or the correlation functions of the 1D model and data profiles). One therefore has to identify the correct pose from these candidates. The quality of each candidate pose is assessed first by instantiating the object model from the pose and then by measuring the goodness-of-fit between the model projection and the image data [15]. The goodness-of-fit is indicated by a scalar evaluation score. The higher the score, the better the pose.

After all candidate poses have been evaluated, they can be ranked according to their evaluation scores. The candidate pose which yields the highest score is taken as the correct pose. This procedure has proven to be both effective and efficient [9].

Further details on pose evaluation and discrimination may be found elsewhere [15, 2, 9 -10].

6 Model discrimination

So far it has been assumed that we know the identity of the object in the given image (region). This is often not the case, and we have to identify not only the pose but also the class of the object. In our work, the following strategy is adopted.

Each candidate model is considered in turn and the best corresponding pose is identified using the procedure outlined in the preceding section. The evaluation scores of the best poses obtained with all candidate models are then compared. The model which produces the highest score identifies the class of the object.

An example is given in Fig.7 to demonstrate the effectiveness of the strategy. For the vehicle (a high-roof van) in the ROI shown in Fig.7(a), three different

Fig. 7. Vehicle discrimination: a ROI of 350x280 pixels (a), and recovered poses and evaluation scores using (b) High-roof Van (correct), (c) Minibus, and (d) Saloon models.

models were considered: a high-roof van model, a minibus model, and a saloon model. The best poses found for the three models are illustrated in Fig.7(b)-(d) respectively. The corresponding evaluation scores obtained are 9.52 (high-roof van), 6.90 (minibus), and 5.33 (saloon). The score achieved by the correct model (the high-roof van model) is much higher than those by the other two models. Cross-over discrimination trials have also been carried out which confirm the validity of the strategy.

7 Remarks

There are several factors which may affect the performance of the algorithms. The key to the simplicity of the algorithms is the GPC. If the constraint is violated (e.g., due to significant rises and/or dips on road surfaces), the algorithms are likely

to fail. However, the algorithms seem to cope well with roads with a moderate level of rises and/or dips such as that shown in Fig.2. From the theoretical point of view, the unevenness of a road is equivalent to the variations of the extrinsic camera parameters (in particular the rotation parameters) across the scene. Since the variations of camera parameters are easier to model than rises and/or dips of roads, experiments were carried out to study how the performance of the algorithms might be affected by camera parameter errors. Preliminary results suggest that the algorithms can tolerate up to 25% errors in camera parameters without much degradation in performance.

The model independence of the orientation recovery algorithm of Section 4.1 is the result of the key observation that the appearance of most vehicles consists of predominantly two sets of parallel horizontal lines - one along the width direction and one along the length direction. However the algorithm does not seem to require strictly parallel and horizontal lines. It copes well with common road vehicles with curved bodies such as those shown in Figures 4 and 7. It also appears capable of locating vehicles with rounded or streamlined bodies. An example of such cases is given in Fig.8, where the pose of the VW Beetle is well recovered.

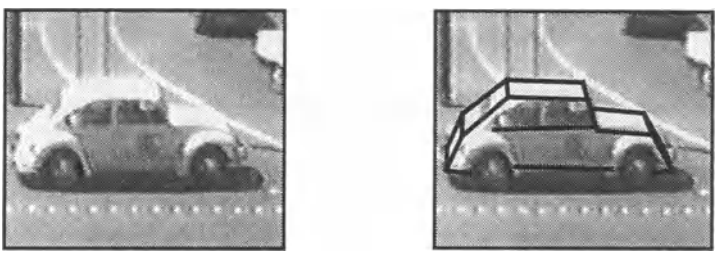


Fig. 8. An example of locating vehicles with rounded or streamlined bodies.

8 Conclusions

We have outlined our recent work on vehicle localisation and discrimination as part of the model-based traffic vision system developed at the University of Reading. Emphasis has been on locating vehicles in given regions of interest. Two groups of algorithms have been described, one based on image line segments, and the other on image gradients. The algorithms exploit several known physical and geometric constraints in road traffic scenes to reduce computational cost.

The algorithms have been tested extensively with routine outdoor traffic images. Successful results have been included to demonstrate the effectiveness of the algorithms.

Acknowledgement

This work was carried out as part of several research projects funded by CEC (under ESPRIT project P.2152), DTI (under grant TIO-68), and EPSRC/SERC (under grants GR-H/90278 and GR-J86322).

References

1. M. Kilger, A shadow handler in a video-based real-time traffic monitoring system, Proc. of IEEE Workshop on Applications of Comput. Vision, pp.1060-1066, Palm Spring, CA, 1992.
2. G. D. Sullivan, Visual interpretation of known objects in constrained scenes, Phil. Trans. Roy. Soc. (B), vol.337, pp.361-370., 1992.
3. E. D. Dickmanns and B. D. Mysliwetz, Recursive 3-D road and relative ego-state recognition, IEEE Trans. PAMI, vol.14, pp.199-213, 1992.
4. H. Kollnig and H.-H. Nagel, 3D pose estimation by fitting image gradients directly to polyhedral models, Proc. 5th ICCV, pp.569-574, MIT, MA, 1995.
5. T. N. Tan, Applications of computer vision in road traffic analysis, Proc. of 2nd CAST Congress of Young Scientists, pp.117-124, Beijing, China, 1995.
6. J. Malik *et. al.*, Smart cars and smart roads, Proc. of 6th BMVC, pp.367-381, Birmingham, England, 1995.
7. A. D. Worrall, G. D. Sullivan and K. D. Baker, Advances in model-based traffic vision, Proc. of 4th BMVC, pp.559-568, Surrey, England, 1993.
8. T. N. Tan, G. D. Sullivan and K. D. Baker, Structure from motion using the ground plane constraint, Proc. of 2nd ECCV, pp.277-281, Italy, 1992.
9. T. N. Tan, G. D. Sullivan and K. D. Baker, Recognising objects on the ground-plane, Image and Vision Computing, vol.12, pp.164-172, 1994.
10. T. N. Tan, G. D. Sullivan and K. D. Baker, Pose determination and recognition of vehicles in traffic scenes, Proc. of 3rd ECCV, pp.501-506, Sweden, 1994.
11. T. N. Tan, G. D. Sullivan and K. D. Baker, Fast vehicle localisation and recognition without line extraction, Proc. of 5th BMVC, pp.85-94, York, England, 1994.
12. T. N. Tan, G. D. Sullivan and K. D. Baker, Fast algorithms for object orientation determination, Proc. of SPIE, vol.2488, pp.263-273, 1995.
13. M. Dhome *et. al.*, Determination of the attitude of 3-D objects from a single perspective view, IEEE Trans. PAMI, vol.11, pp.1265-1278, 1989.
14. T. N. Tan, G. D. Sullivan and K. D. Baker, Line-based object scale and pose estimation, Proc. of ACCV93, pp.347-350, Osaka, Japan, 1993.
15. K. S. Brisdon, Hypothesis verification using iconic matching, PhD thesis, The University of Reading, 1990.
16. Y. C. Liu *et. al.*, Determination of camera location from 2-D and 3-D line and point correspondences, IEEE Trans. PAMI, vol.12, pp.28-37, 1990.
17. T. N. Tan, G. D. Sullivan and K. D. Baker, Linear algorithms for object pose estimation, Proc. of 3rd BMVC, pp.600-609, Leeds, England, 1992.

Contenet-Based Retrieval

SWIM:
A Prototype Environment for Visual Media Retrieval

HongJiang Zhang
Institute of Systems Science, National University of Singapore
Heng Mui Keng Terrace, Kent Ridge, Singapore 0511
Republic of Singapore
zhj@iss.nus.sg

Abstract

This paper presents our work in developing a system, SWIM (Show What I Mean), an prototyping environment for development of content-based image and video retrieval tools. A variety of features for representing image and video content implemented in SWIM has been presented. A set of functional modules of SWIM with unique features, including visual query formation, video parsing and abstraction, image and video retrieval, and video browsing, has been presented with example of their applications.

1. Introduction

With rapid advances in communication and multimedia computing technologies, accessing mass amounts of multimedia data is becoming a reality. However, it is definitely not enough simply to store and display images and video as in conventional databases with text-based approach for representation and index of visual data. New technologies are needed to organize and search these vast amount of data in a content based manner to facilitate effective use of these data. We believe that such tools should be based on image and video indexing and retrieval schemes that make use of properties inherent in the visual data themselves. The idea behind this is that the natural way to retrieve *visual* data is by a query based on the *visual* content of an image: colors, textures, and shapes of image objects, and related layout and location information.

Search based on visual content requires a paradigm which differs significantly from that of both traditional databases and image understanding systems. If our aim is to develop new tools to manage and search visual data in a database environment in a more effective and content-based manner, the key issues which has to be addressed can be summarized as below[4]:

- selection, derivation, and computation of image features and objects that provide useful query expressiveness;
- retrieval methods based on similarity, as opposed to exact matching;

- user interface that supports the visual expression of queries and allows query refinement and navigation of results; and
- effective index schemes compatible with the expressiveness of the queries.

In this paper, we will address these issues by presenting our work in developing a system, SWIM (Show What I Mean), an prototyping environment for development of content-based image and video retrieval tools. This is a part of an on-going effort to develop power tools for intelligent multimedia indexing and retrieval. **Figure 1** show a system architecture of SWIM, at the heart of which are the *feature extraction* and *feature comparison* modules. The extraction module contains preprocessing algorithms that extract their content features from both input and query image/video. The comparison module performs similarity filtering based on similarity metrics. The two modules are supported by the *browsing and feedback* module and the *interactive query formation* module, to perform content based retrieval and browsing of visual data.

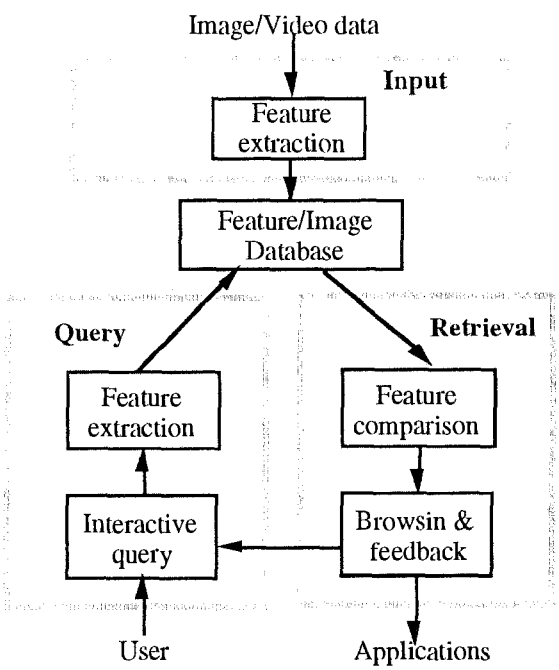


Figure 1: System architecture of SWIM

In Section 2 of this paper, we present the primitive visual properties which we have evaluated and integrated into the system. To handle video data, a video parsing and abstraction sub-system has been developed, which will segment video into its basic indexing units–shots and extract key frame to represent the content of the video in an abstracted manner. Section 3 present tools for query formation,

retrieval and browsing tools of the system and examples of retrieval result based on visual queries. Finally, Section 5 summarizes out current efforts and research issues.

2. Content Representation: Image and Video

As discussed in Section 1, the first issue in developing visual content based indexing and retrieval schemes is to identify and compute *representation primitives*, based on which the content of image can be indexed, compared, and classified. Since the ideal solution of using semantic primitives is not feasible, our representation primitives are based on information accessible low level features. To handle both still images and video, we have developed and implemented two types of representation: still image based primitives and video sequence based primitives. Also, we apply a general-purpose approach, which makes no assumptions at all about the nature of the images being searched.

2.1 Image content representation features

2.1.1 Color features

Color has excellent discrimination power in image retrieval systems. Our representation primitives for color features include color histogram, dominant colors, and statistical moments.

Color histogram: Color histograms are invariant under translation and rotation about the view axis and change only slowly under change of angle of view, change in scale, and occlusion [10]. There are three major issues in using color histogram to represent image content: choosing of color space, color space quantization, and similarity metrics.

We have implemented histogram based retrieval algorithms in 3 color spaces: RGB, opponent and Munsell; and it is observed that RGB gives the poorest performance. Though algorithms using histograms defined in other color spaces are available in SWIM, we have chosen the Munsell space as the main one because it is close to human perception of colors [8]. As in QBIC, we have quantized the color space into 64 "super-cells" using a standard minimum sum of squares clustering algorithm[4]. A 64-bin color histogram is then calculated for each image where each bin is assigned the normalized count of the number of pixels that fall in its corresponding supercell.

The similarity between two color histograms, I and Q, each consisting of N bins, is quantified by the following metric:

$$D_{his}^2(I,Q) = \sum_i^N \sum_j^N a_{ij}(I_i - Q_i)(I_j - Q_j) \tag{1}$$

where the matrix a_{ij} represents the similarity between the colors corresponding to bins i and j, respectively. This matrix needs to be determined from human visual perception studies, and we have derived it using the method of [8]. Notice that if

a_{ij} is the identity matrix, then this measure becomes Euclidean distance. We have also implemented histograms *intersection* operation as similarity comparison, though as a similarity measure, it does not cater for the similarity between colors in neighboring bins[10].

Dominant colors: Because, in most of images, a small number of color ranges capture the majority of pixels, these dominant colors can be used to construct an approximate representation of color distribution. Therefore, only the largest bins need be selected as the representation of any histogram; and a partial histogram comparison may be computed over this reduced set of bins. Experiments have shown that using only a few dominant colors will not degrade the performance of color image matching [10, 5].

Color moments: Because a probability distribution is uniquely characterized by its moments, we represent a color distribution by its first three moments[9]:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} p_{ij} \tag{2}$$

$$\sigma_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - \mu_i)^2)^{\frac{1}{2}} \tag{3}$$

$$s_i = (\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - \mu_i)^3)^{\frac{1}{3}} \tag{4}$$

where p_{ij} is the value of the i-th color component of the j-th image pixel. μ_i, defines the average intensity of each color component; and σ_i and s_i, respectively, define the variance and skewness. A weighted Euclidean distance is used as the similarity measure for color moment based image search.

2.1.2 Texture features

Texture has long been recognized as being as important a property of images as is color, if not more so, since textural information can be conveyed as readily with gray-level images as it can in color. Among many alternatives we have chosen two models which are both popular and effective in image retrieval: Tamura features [11] and the Simultaneous Auto-Regressive (SAR) model [7].

Tamura Features: The Tamura features are *contrast, directionality*, and *coarseness,* which were introduced as quantification of psychological attributes. Coarseness is a measure of the granularity of the texture and it is derived from moving averages computed over windows of different sizes. Directionality is computed from distributions of magnitude and direction of gradient at all pixels. The quantification of contrast is based on the statistical distribution of pixel intensities [11].

SAR Model: Given an image of gray-level pixels, the SAR model provides a description of each pixel in terms of its neighboring pixels. Our implementation uses the multiresolution SAR (MRSAR) model [7] which describes textures at

different resolutions in order to model different granularities. Images are represented by a multiresolution Gaussian pyramid obtained by low-pass filtering and subsampling operators applied at several successive levels. We have chosen four levels of resolution, and the collection of model parameters at each level is used as the texture features of each image.

Similarity: For either model texture is represented quantitatively as a feature vector X. For the SAR model the distance between two such vectors is given by the Mahalanobis function:

$$D_{Mahal} = \left(X^1 - X^2\right)^T C^{-1}\left(X^1 - X^2\right) \tag{5}$$

C is the covariance matrix which models pair-wise relationships among the individual model features. Because the Tamura features are almost uncorrelated, the following simplified Mahalanobis function may be used instead:

$$D_{simp} = \sum_{i=1}^{J} \frac{\left(x_j^1 - x_j^2\right)^2}{c_j} \tag{6}$$

2.1.3 Shape features

Dominant objects in images represent important semantic content and are best represented by their shapes. In our implementation dominant objects are obtained by an interactive outlining algorithm [2]. Cumulative turning angles are used as the similarity metric, because they provide a measure closer to human perception of shapes than algebraic moments or parameteric curves, and they are invariant under translation, rotation, change of scale, and with respect to convex and non-convex polygons.

2.2. Video content representation

2.2.1 Video parsing

There is two more steps in obtaining video content representation compared with still images: a *temporal segmentation* process that segments a video program into elemental indexing units corresponding to *camera shots*; and a *video abstraction* process that maps an shot to a small number of representatives, called *key frames*, which best represent the content of the video sequence in an abstracted manner. Image features can then be applied directly to the key frames to represent the shot content. In addition, shot based features, mainly derived from motion features, will composite another set of content features for video.

To accomplish these, we have develop a set of robust video segmentation and abstraction algorithms as a function module for video data in the input module of SWIM system. The video parsing module detects not only simple camera breaks but also gradual transitions implemented by special effects[12]. Also, the parsing module can perform segmentation and key frame extraction using compressed data directly[13].

The challenge in extraction of key frames is that it needs to be only automatic but also content based so that they maintain the important content of the video while remove all redundancy. Our unique and robust key frame extraction technique utilizes low-level image features and motion information resulting from the parsing process. Certain heuristics are also applied, such as the decision to use the first frame of every shot as a key frame. In addition users can adjust parameters to control the density of key frames in each shot [14].

2.2.2 Shot based visual features

Key frame based representation utilizes only spatial information of video frames and ignore the temporal nature of a video to a large extent. With such a representation only it will be difficult to support event-based classification and retrieval. Therefore, apart from key frame based features, we have also been studying temporal features for more shot or event based representation of video content.

Motion features: We base our motion representation on statistical motion features, rather than object trajectory based as in some other works. These features include directional distributions of motion vectors and average speed in different directions and areas., derived from optical flow between frames.

Other features: Detection of camera operations, including panning, tilting, and zooming, can be achieved by analyzing either motion fields or spatio-temporal images, and provides another set of data for shot representation. Temporal variation may also be represented by the mean and covariance of average brightness, a few dominant colors calculated over all frames in a shot.

3. Query, Retrieval and Browsing

An appropriate representation of image and video content based on visual feature provide a foundation for content-based indexing and retrieval. In this section, we present a set of such tools in SWIM system based on the representation features presented in the last section.

3.1 Query formation

In building SWIM system, we believe that queries are most likely to be effective if they can be specified *visually*, *interactively*, and *iteratively*; and the same is true for assessing retrieval results. This is understandable since visual description can be understood faster than text description of visa data. Also, only with visual feedback can we exploit the characteristics of retrieved images to refine a query for subsequent retrievals. Based on these considerations, *query by visual examples* is the key method for image and video query. In SWIM, tools are provided to composite visual examples by template manipulation, painting and selecting from previous query results. In case of video retrieval, a video shot, instead of a still image, can also used as a query example to retrieval similar video shots.

3.2 Content-based retrieval of images

Once an example image is presented as a query, its visual features will be calculated and search will be based the similarity between the query example and images in the database. The retrieval system we are building behaves more like an *information filter* based on similarity ranking.

An example of query by example is shown in **Figure 2**: This query was aiming at retrieving all images which contain green gardens similar to the query image. Color histogram matching is used in the retrieval, and 10 retrieved images are listed in descending order of similarity to the example image. In our database there are actually 13 images similar to the example image, all of which were retrieved; but the interface only allows 10 images to be displayed.

It is noted that the retrieval interface provide a function that allow the user to pick an image in the retrieved list as an example to launch an query iteratively: a feedback to narrow or refine any queries that are given..

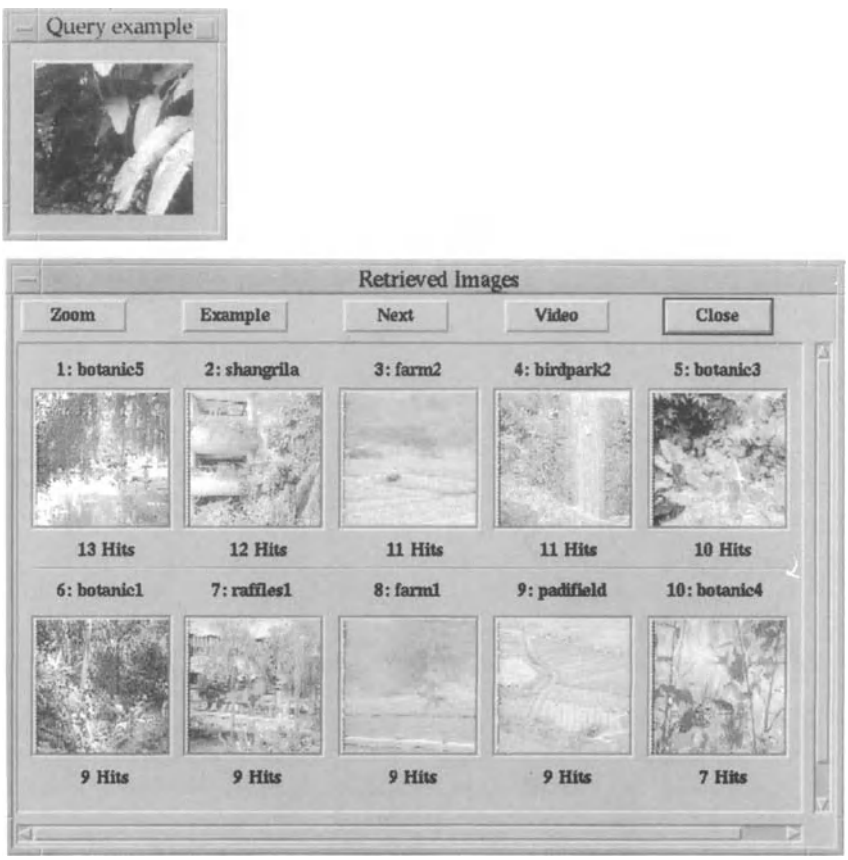


Figure 2: An example of visual example based query and its result.

3.3. Content-based retrieval of video

Similarly, retrieval tools as shown in **Figure 1** can be applied to retrieve video sequences represented by their key frames. In this case, once a key frame has been retrieved, the user may view its associated video clip by clicking the "Video" button in the "Retrieved Images" window. This will initiates a video player, which is cued by the location of the key frame. That frame may also be used to derive a hierarchical display, as will be seen in *Figure 3*. On the other hand a query can also be initiated from browsing.

To support shot based query, we are also developing tools for retrieval of video based on shot examples and motion features. Queries such as "find all shots with a camera panning at 10°/second" can be easily satisfied based on temporal representations. We can also combine shot and key frame features in a single query.

3.4. Content-based browsing tools for video

Another unique feature in SWIM system is its content browsing tools for video, which utilizes the content information obtained from video parsing, including segment boundaries, camera operations, and key frames. Our browsing tools support two different approaches to accessing video source data: *sequential* access and *random* access; and accommodate two levels of granularity—overview and detail—along with an effective bridge between the two levels.

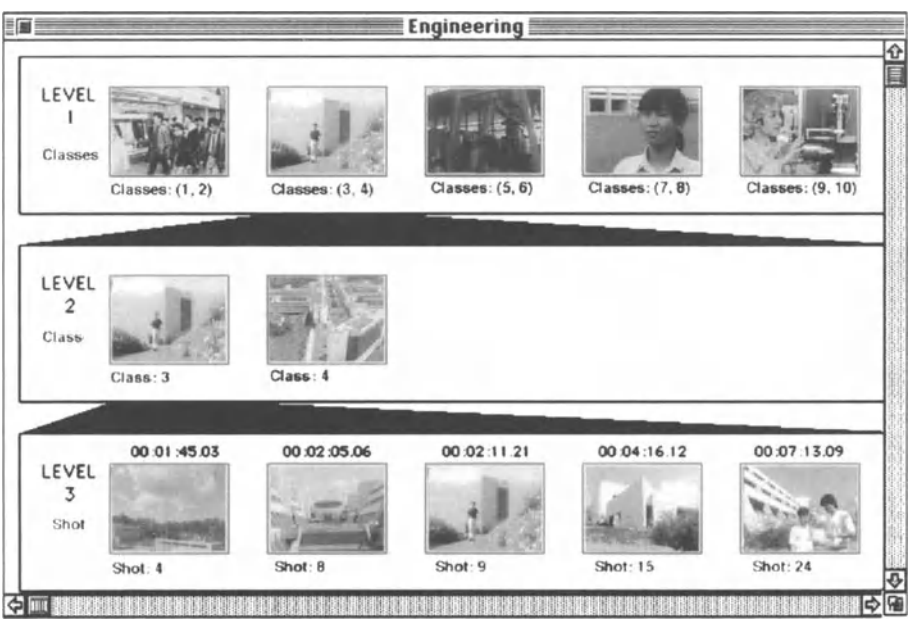


Figure 3: Cluster-based hierarchical video browsing

In our system, sequential access browsing takes place through a VCR-like interface. Overview granularity is achieved by playing *only* the extracted key-frames at a selected rate. Detailed granularity is provided by normal viewing, with frame-by-frame single stepping. **Figure 3** shows a hierarchical browser designed to provide random access to any point in a given video: a video sequence is spread in space and represented by frame icons which function rather like a light table of slides. At the top of the hierarchy, a whole video is represented by five key frames, each corresponding to a class of shots. As we descend through the hierarchy, our attention focuses on smaller groups of shots, single shots, the representative frames of a specific shot, and finally a sequence of key frames.

The hierarchical browser presented above is *class-based* such that at the top of the hierarchy is 5 classes of shots, clustered according to their visual content. To support this feature, two types of algorithms, ISODATA partitioning and hierarchical clustering have been developed. When a list of video clips are provided, the parsing system will use either key frame and/or shot features to cluster the shots into classes of similar content. As a result, user can know roughly the content of each calls of shots even without moving down to lower level of the hierarchy. Such clustering is also very useful in index building and computer-assisted video content annotation.

4. Concluding Remarks: Research Issues

In this paper we have presented our work in developing SWIM, an prototyping environment for content based image and video indexing, retrieval and browsing. The core of this system is its use of low-level visual features as a representation of image and video content, its automatic video parsing and abstraction algorithm and its content based browsing tools. Such a system and its function modules can be applied in many applications, including multimedia databases, video on demand and content-based video compression.

As an active research area, there are still many issues to be addressed in content based image and video indexing and retrieval. Some of the issues we are studying include:

- To bridge between low-level features to semantic content of visual data, especially video, in certain application domain. Perceptual similarity measures based on low level feature are needed.
- Event based and spatio-temporal features for representation of video content.
- Application issues. We need to identify applications that will utilize content technologies and pose more specific requirement.
- Fusion of information derived from audio, speech and text media in order to understand video content and to realize real content based retrieval. Only with such an integration, can applications like news video parsing and news on demand can be automated to a great extent.

Acknowledgment

The work presented in this paper is a team work and the author would like to thank the team members for their contribution: Chien Yong Low, Jian Hua Wu, Di Zhong, Stephen W. Smoliar and Qi Tian.

References

1. E. M. Arkin *et al.*, An Efficiently Computable Metric for Comparing Polygonal Shapes, *IEEE Transactions on Patt. Analy. and Mach. Intell.*, 13 (3), pp. 209-216, 1991.
2. D. Daneels *et al.*, Interactive Outlining: An Improved Approach Using Active Geometry Features, *Proc. IS&T/SPIE. Conf. on Storage and Retrieval for Image and Video Databases II*, San Jose, CA, 1993.
3. R. Duda and P. Hart, *Pattern recognition and scene analysis*, Wiley, New York, 1973.
4. C. Faloutsos *et al.*, Efficient and Effective Querying by Image Content, *Journal of Intelligent Information Systems* 3, pp. 231-262, 1994.
5. Y. Gong *et al.*, An Image Database System with Content Capturing and Fast Image Indexing Abilities, *Proc. International Conference on Multimedia Computing and Systems*, Boston, MA, 1994, pp. 121-130.
6. T. Kato *et al.*, A Sketch Retrieval Method for Full Color Image Database: Query by Visual Example, *Proc. 11th International Conference on Pattern Recognition*, Amsterdam, HOLLAND, 1992, pp. 530-533.
7. J. Mao and A. K. Jain, Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models, *Pattern Recognition*, 25 (2), pp.173-188, 1992.
8. M. Miyahara, and Y. Yoshida, Mathematical Transform of (R,G,B) Color Data to Munsell (H,V,C) Color Data, *Proc. of SPIE Visual Communication and Image Processing*, 1001, pp. 650-657, 1988.
9. M. Stricker and M. Orengo, Similarity of Color Images, *Proc. IS&T/SPIE. Conf. on Storage and Retrieval for Image and Video Databases III*, San Jose, CA, 1995.
10. M. J. Swain and D. H. Balllard, Color Indexing, International Journal of Computer Vision, Vol.7, pp.11-32, 1991.
11. H. Tamura, S. Mori, and T. Yamawaki, Texture Features Corresponding to Visual Perception, *IEEE Trans. on Syst., Man, and Cybern.*, 6 (4), pp.460-473, 1979.
12. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, Automatic Partitioning of Full-motion Video, *Multimedia Systems* 1 (1), pp. 10-28, 1993.
13. H. J. Zhang *et al.*, Video Parsing Using Compressed Data, *Proc. IS&T/SPIE Conf. on Image and Video Processing II*, San Jose, CA, 1994, pp. 142-149.
14. H. J. Zhang, S. W. Smoliar, and J. H. Wu, Content-Based Video Browsing Tools, *Proc. IS&T/SPIE Conf. on Multimedia Computing and Networking 1995*, San Jose, CA, 1995.

Video Segment Indexing Through Classification and Interactive View-Based Query

John Chung-Mong Lee†, Wei Xiong†, Ding-Gang Shen‡and Ruihua Ma†

†Department of Computer Science
The Hong Kong University of Science & Technology, Hong Kong
‡Institute of Optic-Fibre Technology
Shanghai Jiaotong University, Shanghai, China
Email: {cmlee,csxwei,rhma}@cs.ust.hk

Abstract. As video information proliferates, managing video sources becomes increasingly important. Indices must be constructed to allow any future retrieval. We distinguish two categories of indexing: (i) those that are general-purpose and do not make use of domain-specific knowledge, and (ii) those that are application-dependent. In this paper, we present our work in both categories within the VideoBook project. We discuss how to structure video data into shots (physical parts) and clusters (semantic parts). A video partitioning algorithm is described. Its effectiveness and efficiency lies in the use of both statistical and spatial information in the images without, however, having to examine the entire images. To improve the querying efficiency, we propose to investigate in two directions: deriving higher-level indices through classification and providing a method that finds targets of interest through interactive learning. The first technique takes advantage of domain knowledge of underlying applications. The second technique accounts for quantification effect and noise in images and accommodates "learning from negative examples", resulting into quite good discriminating power. Experimental results are given to demonstrate the effectiveness of our approach.

1 Introduction

With the rapid progress in video technology, large amounts of video sources become available. This availability is not synonymous with *accessibility*. As a matter of fact, traditional text-based methods for video management do not allow easy access to the video sources. The reason for this is twofold. First, video data is not structured, i.e., they have few or no temporal tags. Secondly, indices are basically textual and not rich enough because of the cost required to index video sources. Thus, generally speaking, the solution to this problem lies in structuring video data and associating it with much richer indices. In particular, visual indices may allow a user to access desired video segments by directly

* This research was supported by Sino Software Research Center of the Hong Kong University of Science & Technology.

making use of visual cues. Most prototype systems proposed in the literature, including ours, follow this direction [2, 6, 12, 7].

Structuring video data that consists of segmenting the continuous frame stream into physically discontinuous units, generally called *shots*, is a basic operation. In general, these physical units need to be clustered to form more semantically significant units, such as scenes. This so-called *story-based* video structuring has been used in video information browsing systems (*eg.* [8, 4]). The shots or scenes are described by one or several representative frames, known by the name *key frames* [8, 12]. At the basis of all these are camera break detection and key frame selection. Camera breaks are usually characterized by brusque intensity pattern change between consecutive frames at the boundary. While camera break detection involves only the determination of a threshold, key frame selection is event-driven, which implies that it is subjective and/or context-dependent. Structured video data should be attached with visual indices to allow visual content based retrieval. Thus it comes the problem of knowing what features to use, how to extract them from images, and how to use them for indexing – a key problem in video retrieval. Indices can be of several different levels. Currently, most low-level features found in image processing or computer vision have been attempted as indices. Such features include color, texture, shape, sketch and motion. Success in these areas, however, is not so impressive. In fact, the ways in which a feature is exploited vary from system to system, leading to different effectiveness.

When used as indices for retrieval, low-level features are not very efficient. They act most of the time as constraints in filtering. Also their use implies on-line computations, which tend to be prohibitive whenever the search space becomes big. Higher level features, up to symbolic ones, on the other hand, are quite efficient in retrieval. For example, it is not easy to find scenes containing a dog if one uses features such as color, shape or the like; but it is quite an easy thing if at the indexing stage the scenes are annotated with the term dog, either manually or by an algorithm. It is also well known that the extraction of higher-level features is as difficult as they are efficient. It is, however, sometimes feasible when we make careful use of available context knowledge.

In this paper, we report techniques recently developed at HKUST relating to VideoBook, a video database management system. We first present video structuring (Section 2), especially camera break detection. We also discuss how to do video structuring through semantic clustering. In Section 3, we address the use of domain knowledge for shot classification. In Section 4, we show how to improve querying efficiency through interactive learning. In each of the above sections, we give algorithms and experimental results. Finally, we conclude and discuss future research directions in Section 5.

2 Video Structuring

2.1 Video partitioning using Net Comparison

The partitioning process consists of the detection of boundaries between uninterrupted segments (camera shots) which involve screen time, space or graphic configurations. These boundaries, also known as transitions, can be classified into two categories: gradual and instantaneous. The most common transition is camera breaks.

Several methods such as pairwise comparison, likelihood comparison and histogram comparison have been introduced [5, 10]. These methods have their merits and limitations. The histogram comparison method is insensitive to image movements since it considers intensity/color distribution – a statistical entity – between consecutive images. But it fails if the intensity/color distributions are similar because it ignores spatial information [3]. Both pairwise comparison and likelihood comparison make use of spatial information but the former is too sensitive to image movements and easily causes false alarms, whereas the latter suffers from computational complexity. To overcome these problems and further reduce the computation time, we propose a method, called Net Comparison (NC). It takes advantage of the robustness of the histogram method and the simplicity of the pairwise method by comparing a statistical quantity – the mean value of intensity – along the predefined net lines. Thus only part of the image is inspected.

The algorithm works as follows. First, M points, uniformly distributed over an image, are chosen. Around each of the points a non-overlapped rectangular window is taken, and its mean intensity value $\bar{I}_m$ $(m = 1..M)$ is computed. A camera break is declared if the total number of changed regions is larger than a threshold N_{CR}. A region m is said to have changed if

$$|\bar{I}_m^t - \bar{I}_m^{t+\Delta t}| > \Delta I$$

where the superscripts denote two successive frames and ΔI is a predefined threshold.

For the purpose of comparison, we have implemented four other methods for camera break detection: pairwise, likelihood, global histogram, and local histogram. Many experiments have been conducted, on both color and black/white video. It turns out that the proposed method outperforms the others both in accuracy and in speed (for details, see [9]).

2.2 Clusters and key frames

A continuous video is segmented into shots by partitioning. Each shot is represented by or abstracted into one or more frames, commonly called *key frames*. Key frames can serve two purposes: *browsing* and *computation of indices*. In retrieval by browsing, showing shots using key frames as they are may confuse an untrained user rather than help him get the *story*, hence facilitate the retrieval. In light of the table of contents of a book, shots must be organized into several

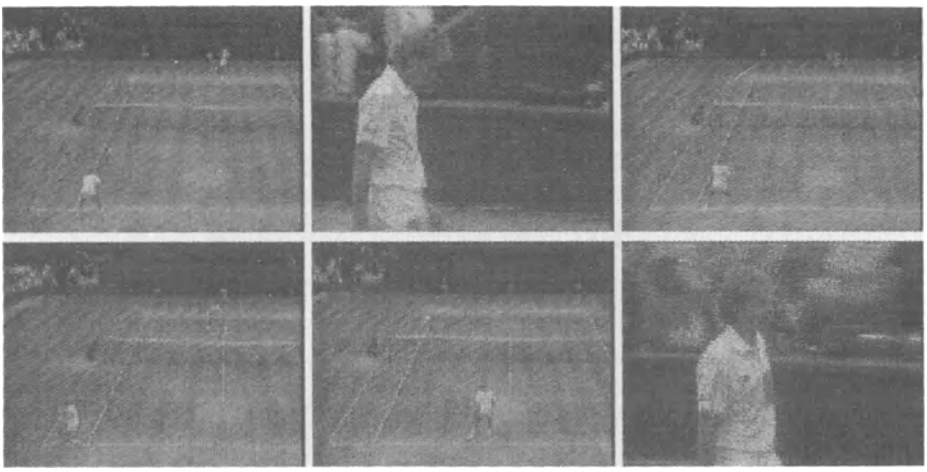


Fig. 1. Tennis competition shots.

levels of semantic abstraction. We call this *conceptual* or *semantic clustering*. The unit is *cluster*, which is a collection of semantically related shots and/or clusters [4, 1]. Now the question raised is how to find clusters, and to a less extent, how to choose key frames to represent them.

Generally speaking, such a semantic clustering can only be accomplished by a human operator. Nonetheless, in some particular circumstances, automatic clustering is still possible. Section 3 presents an example in which football shooting shots are clustered using the fact that shootings must happen near the goal posts and goal posts can be detected reliably. In [11], Zhang *et al.* use both spatial and temporal structure to classify anchor-person shots and episodes in a news video. Anchor-person shots identification makes use of *persistancy*, which can be applied to other cases. For example, in a conversation sequence, there are limited fixed viewpoints, although shots change frequently. Thus a sequence of this kind may be well summarized using only a few key frames. Another example deals with sport competitions. A coach may be interested only in shots of a competition, not of the public. A shot classification algorithm may be devised (using color histogram comparison, for example) which retains only shots containing the court (Fig.1).

3 Scene Classification for Indexing

As discussed in introduction, low-level image features are not very efficient for retrieval, because they are general, that is, not very discriminant. Further, all queries cannot be constructed on the basis of these indices. On the contrary, indices of higher-levels are much more efficient. Therefore we should provide this

kind of indices as much as possible. However, their derivation poses problems. Manual input is quite time-consuming and hence impractical. Current computer vision technology is not able to do general scene classification or interpretation. The alternative is thus to take advantage of application context whenever this is possible. This will considerably improve retrieval efficiency.

As an example, we have studied the case of football, one of the most popular sports of the time. Shooting is often the most important instant or the moment that people are most interested in, in a football match. Locating such instants may help a sport programme editor to rapidly make a summary of a match or find out all shootings. One way to do this is to identify all segments containing the penalty areas, or more simply, the goal post, because when shootings occur, the goal post is in general in the camera's view. Now we will show how the specific context knowledge allows us to devise a simple algorithm to perform our task.

3.1 Football Goal Detection

A goal post is composed of two vertical bars and a horizontal bar. Due to viewpoint differences, the horizontal bar is seen as slanted most of the time. On the contrary, no matter what the viewpoint is, the two vertical bars are almost always vertical in the image plane. Thus we model the goal post as (two) vertical bar(s) in the image. One can think of its detection as a simple edge detection and linking problem. In fact, problems due to low resolution, lighting conditions, as well as various background noise (e.g., advertisements), must be taken into account.

The goal posts are painted in white. In the image, a vertical bar is a line (i) composed of flat roof edge points, (ii) of some length, and (iii) of course, vertical. The following procedure detects potential goal post elements.

for each pixel (i, j) in the image
  if (i, j) is a roof edge
  then mark (i, j) as potential bar element
    go to $(i, j + 1)$.

To determine if a (i, j) is a roof edge, we require that there be simultaneously a up-going jump on the left of (i, j) and a down-going jump on the right, i.e.,

$$I(i, j) - I(i - k, j) > \Delta I \text{ and}$$
$$I(i, j) - I(i + l, j) > \Delta I, 0 < k, l < \Delta W$$

where ΔI is a threshold and ΔW is a predefined width of 1 or 2 pixels.

All edges so detected are not bar elements. Such edges form a vertical line of some length after being linked together. At the same time, there may be bar elements that are missed due to noise or lighting conditions associated with a vertical bar. Such points are recovered by examining if there are potential bar elements right above and under them. Further, the spatial resolution limit of one pixel is accounted for by allowing one horizontal pixel displacement. Finally, all

vertical lines the length of which is more than a predefined length (depending on the image size) are declared vertical goal posts.

Our algorithm has been applied to the video tape – "94 World Cup, 50 exciting shootings". It has successfully detected shots that contain goal posts. Fig.2 shows some example images of goal posts detected.

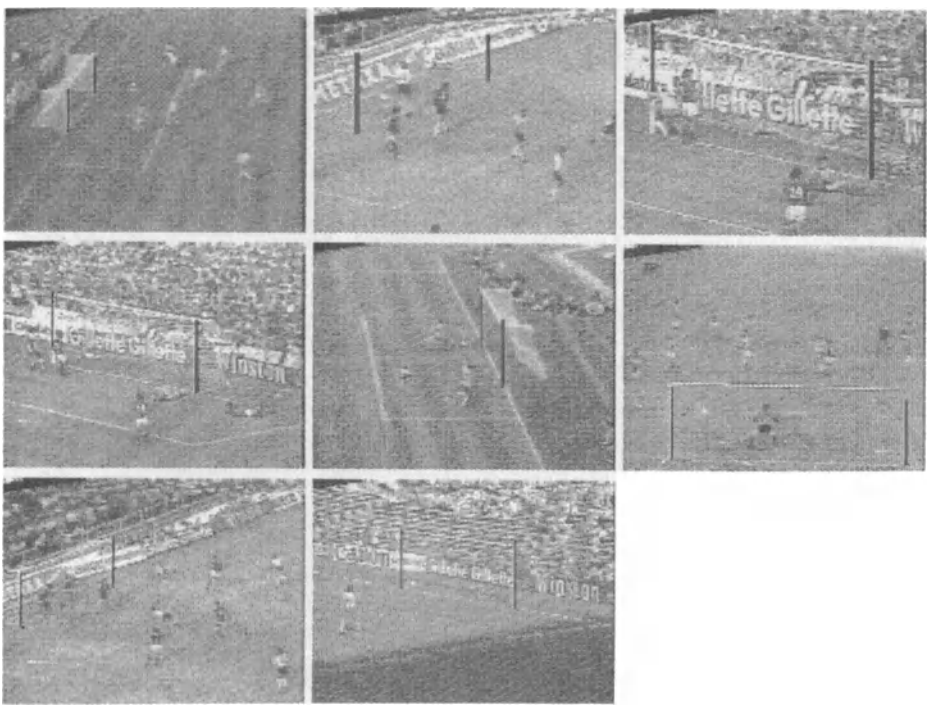


Fig. 2. Localization of vertical bars for goal post detection.

4 Querying through Interactive Learning

For the sake of reducing indexing effort, visual indices will predominantly consist of low-level image features. However, their efficiency for search reduction is limited by their generality. To remedy this, we have proposed to derive symbolic indices using *a priori* knowledge when it is available in Section 3. In this section, we introduce another approach, namely *view-based learning*. It can learn visual features in an effective way and this will allow much more increased discriminating capability. More precisely, instead of using statistical features of the

whole image, we deal with features of user-selected regions or objects. We are interested in locating key frames that contain objects of interest. The outline of this novel approach is as follows.

1. The user chooses one or more key frame(s) which contain objects of interest.
2. The user selects a feature (color, shape, texture, etc.).
3. The user draws a rectangle on the image and tells the system whether the enclosed region is interesting (positive example) or not interesting (negative example). The program learns parameters of the feature from the specified area.
4. The program applies the learned feature parameters on the image.
5. Repeat steps 3 and 4 until the user is satisfied with the result.
6. The program searches for the targets in all of the frames and reports the search result.

Currently, we have implemented the method with color only. The use of other features like shape and texture are under development.

Color segmentation algorithms based on statistical models are basically pixel classification techniques applied to some 3D color space. We adopt the HSV model (Hue, Saturation, Value) to represent and compare the color information. We perceive color as hue, saturation and value (intensity). Hue, H, corresponds to the pure color pigment, saturation S describes the purity of colors (red is highly saturated and pink is unsaturated), and value V contains the relatively bright colors. Because of the perceivable properties in HSV space, it is particularly suitable for the view-based purpose.

In order to account for the quantification effect of the color space as well as slight changes of illumination due to time or viewpoint, we regenerate the learnt color distribution (histogram). The regenerated histogram has the form of a Gaussian distribution function or the sum of such functions with the peaks at the same position(s) as in the learnt one. During learning through negative examples, the value of an entry in the histogram is set to negative whenever the entry appears in a negative example. The resulting histogram is applied to images to classify pixels. This method proves to be robust to small changes in illumination and yet possesses excellent discriminating power. Fig.3 shows some results of our experiments in searching for sofas in the images.

5 Conclusion

In this paper, we have presented our viewpoint on the video data management problem. We consider that the key for efficient retrieval relies on video data structuring and effective indexing. We have described our effort in this direction. Some techniques recently developed in our laboratory are presented. Our video partitioning algorithm (NC) outperforms other existing ones both in accuracy and in speed due to the fact that it uses both statistical and spatial information on the images without, however, having to process the entire image. We have also shown how to take advantage of domain knowledge of underlying

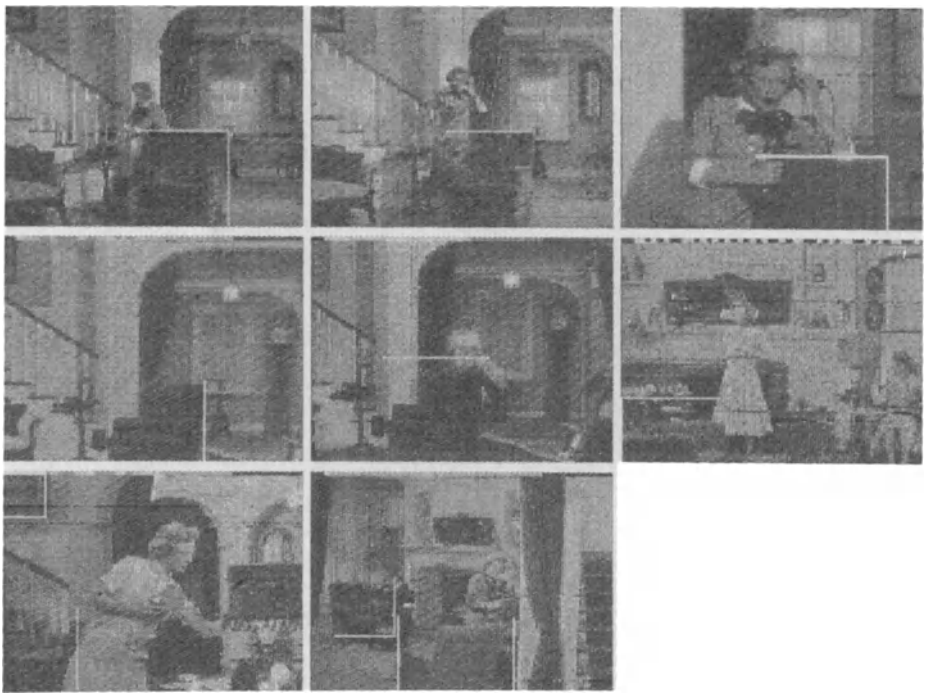


Fig. 3. Object searching with colors.

applications. We have used the knowledge to classify shots allowing derivation of semantic-level indices as well as semantic clustering. We have also proposed a novel approach – interactive learning, to improve the discriminating power of low-level features. The use of one of such features – color, has been studied. The proposed method learns from both positive and negative examples. Furthermore, it accounts for the quantification effect of the color space as well as the slight change of illumination due to time or viewpoint. The resulting algorithm is quite efficient in reducing search space. Experimental results have been provided. For future work, we plan to undertake a thorough investigation into efficient use of low-level image features as well as automatic scene clustering/classification. We will also continue to study interactive learning for retrieval using other features like texture and shape.

References

1. L. S. Huang, C. M. Lee, Q. Li, and W. Xiong. Dynamic object clustering with video database manipulations. Submitted to the IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Databases IV, San Jose, CA, February, 1996.

2. T. Kato. Database architecture for content-based image retrieval. In *SPIE Proc. Image Storage and Retrieval Systems*, volume 1662, pages 112–123, 1992.
3. C. M. Lee and M. C. Ip. A robust approach for camera break detection in color video sequence. In *Proc. IAPR Workshop on Machine Vision Application (MVA'94)*, pages 502–505, Kawasaki, Japan, December 1994.
4. Q. Li and C. M. Lee. Dynamic object clustering for video database manipulations. In *Proc. IFIP 2.6 Working Conference on Visual Database Systems*, pages 125–137, Lausanne, Switzerland, March 1995.
5. Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances. In *IFIP Transactions A-7, Visual Database System II*, pages 113–127, North-Holland, 1992. Elsevier Science Publishers B.V. Edited by E. Knuth and L. M. Wegner.
6. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Query images by content using color, texture and shape. In *SPIE Proc. Storage and retrieval for image and video databases*, volume 1908, pages 173–186, 1993.
7. A. Pentland, R.W. Picard, and S. Scaroff. Photobook: Tools for content-based manipulation of image databases. In *SPIE Proc. Storage and retrieval for image and video databases II*, volume 2185, pages 34–46, 1994. Longer version available as MIT Media Lab Perceptual Computing Technical Report No.255, Nov. 1993.
8. Y. Tonomura. Video handling based one structured information for hypermedia systems. In *Proc. ACM Int'l Conf. on Multimedia Information Systems*, pages 333–344, New York, USA, 1991. ACM Press.
9. Wei Xiong, C. M Lee, and Man Ching Ip. Net Comparisom: A Fast and Effective Method for Classifying Image Sequence. In *IS&T/SPIE Symposium on Storage and Retrieval for Image and Video Databases,San Jose,USA*, 1995.
10. Hong Jiang Zhang and Stephen W. Smoliar Atreyi Kankanhalli. Automatic partitioning of full-motion video. *Multimatia Systems*, 1(1):10–28, 1993.
11. Hong Jiang Zhang, Yihong Gong, S.W. Smoliar, and Shuang Yeo Tan. Automatic parsing of news video. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 45–54, Boston, MA, USA, 15-19 May 1994.
12. Hong Jiang Zhang and S. W. Smoliar. Developing power tools for video indexing and retrieval. In *Proceedings of the SPIE vol.2185 (1994), San Jose, CA, USA, 7-8 Feb. 1994.*, pages 140–149, 1994.

View-Based Techniques for Searching for Objects and Textures

Michael J. Swain, Charles H. Frankel and Mei Lu

Department of Computer Science
The University of Chicago
Chicago, IL 60637 USA
{swain, frankel, ml23}@cs.uchicago.edu

Abstract. We show how it is possible to use the surface properties of objects to find them in image databases. Using color and local spatial cues at multiple resolutions, we can distinguish both textures and individual salient features on the surfaces of objects. Preliminary experiments on databases of size 20 (for the salient features) and 147 (for the textures) show promising results, and suggest we can extend these approaches to large databases.

1 Introduction

We are building a tool, called FINDIT [9], for locating the image of an object from within a large number of images of scenes which may contain the object. The user outlines an object in an image that he wants to find elsewhere in the database (which could be a sequence of images from a videotape, for example), and specifies the constraints on the transformations of the object that are expected to occur. The program works from indexing data structures created off-line and acts as a filter to quickly reduce the possible umber of candidates to a number small enough to be scanned by the user [1]. FINDIT will choose an appropriate search algorithm depending on the selection of constraints by the user. Constraints we intend to explore include constant orientation (with respect to vertical), same view, approximately constant size, no occlusion, constant color of lighting, constant intensity of lighting, and rigidity (i.e. object does not deform).

In FINDIT, the user selects an object or image region, the system develops a representation, and then attempts to find the same object or texture in the image database. This paper describes the features and algorithms that allow us to solve such problems. The same features can be used for finding both textures and objects. The algorithms for finding textures have been tested a large database (147 textures) but have not been tested over changes in view and lighting, although we have reasons to believe that the representations are robust to small changes. The salient feature recognition has been tested over changes in view and lighting, for a database of 20 objects.

2 Texture recognition

The ability to recognize visual textures is critical to recognizing many materials and objects (e.g. types of wood, types of stone, types of animal fur, species of plant etc.).

We describe a technique for recognizing color textures that is designed to find objects distinguishable by their surface properties in large image databases. The representations that we have developed are: compact, capable of supporting search over orientation, scale and changes in lighting, quickly matched, and capable of distinguishing textures from a large database. The approach, called CDFM (Color Distribution and Filter Moments) has been tested on a database containing six thousand subimages from images of 147 different textures obtained from photographs of natural scenes of homogeneous textures; half of these subimages were used for training and half for testing. Once trained, our system was able to correctly classify better than 95% of the subimages in the test set, with 99.8% of the correct textures within the top three matches. We have found statistics of both the color distribution and the responses of steerable filters over multiple scales and color bands to be useful features. The best results were obtained by using both of these sets of features together.

Because the identification of visual textures has been understood to be such a useful skill, much work has been done in the this area. But until recently, the majority of such work tested only on small databases of grayscale textures. Recently there has been growing interest in techniques that will scale up to large databases [5]. Here we describe a technique for recognizing textures that is extremely effective for recognition on a database of close to 150 textures. The use of color information is critical to its success [4, 3].

In the CDFM algorithm, the information extracted is a feature vector, derived from statistics of both the color distribution and the responses of steerable filters over multiple scales and color bands. Each texture is represented by a multi-layer neural network trained on samples of such statistics, with positive examples extracted from the texture and negative examples extracted from the universe of other textures occurring in the database.

2.1 Computation of the Feature Vectors

We included information from both the first and second order statistics of the image. To capture the first order statistics of the textures, we computed first, second, and third moments of the color distribution in three color bands (Y,I,Q), for a total of 9 features [8]. These color bands were chosen to decorrelate the color data. Since one of the color bands (Y) includes only grayscale information, this representation is useful for measuring the usefulness of color information in the representation of textures. The feature vector also includes computed second order statistics in the form of responses from steerable filters (from Freeman's Steerable Pyramid [7]) at three different scales in each color band. The filters in the Steerable Pyramid are composed of four orientations; so there a total of 36 features obtained from the Steerable Pyramid responses. Taken together, the

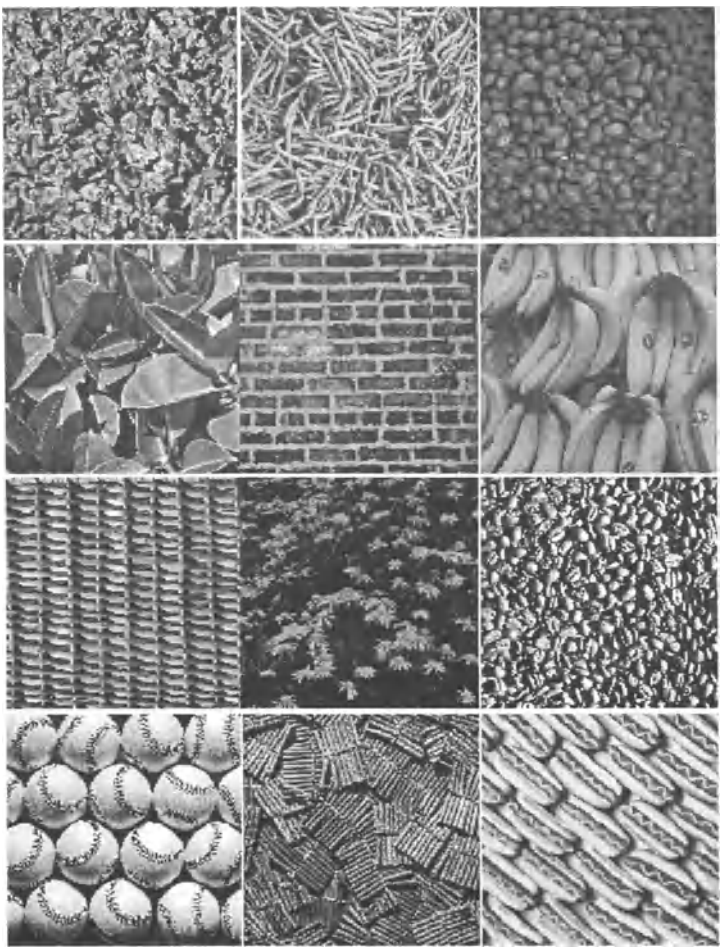


Fig. 1. A selection of the textures in the image database.

moments of the color distribution and the Steerable Pyramid responses add up to a total of 45 features.

We used the distribution of these filter responses in our vector. Which moments of these distribution are appropriate? To increase the efficiency of the recognition process, the vector should be as small as possible. But the vector should still contain sufficient information to distinguish among different images of textures.

In CDFM the first, second and third moments of the (first order) color distribution are utilized. Higher order moments convey progressively less useful information, and so moments higher than the third were omitted as was done in

[8]. Because blurring an image usually has little effect on the first order statistics (see [10]), we gain very little useful information by including the these statistics over different scales. We therefore included the first order statistics only over the smallest (pixel) scale.

For the second order statistics (Steerable Pyramid responses), we used only the standard deviation (second moment) of the distribution. The first moment here does not provide useful information. This is because the mean response of any zero mean filter over an image will be close to zero.

2.2 The Classification System

Our goals in choosing a classifier were that it must learn compact representations of each class and must be able to generate a classification and confidence level for that classification very quickly. Neural networks met all of these requirements.

The units for each axis of our feature vector are non-commensurate. For the images in our datasets, the mean and standard deviation was markedly different for each axis. Under the assumption that the axes of our feature vector contain roughly equal amounts of information we chose, therefore, to normalize the mean and standard deviation of the feature vectors along each dimension before applying them the neural net classifiers. This normalization step proved critical to the success of our classifier.

A separate network was trained for each texture to be recognized. Each neural network was trained on positive and negative examples of the texture; the negative examples were obtained from other textures in the database.

The choice of using a separate neural network for each texture may seem surprising since a single network could provide a more compact representation of the group of textures. Outweighing this possibility is the advantage that smaller networks require less training and suffer fewer convergence problems. Also, using separate networks allows us to add additional textures to the database without retraining the entire classification network from scratch.

2.3 Experiments

In order to test the system, we used textures from three databases: a commercially available CD ROM of textures entitled Having Fun, Vistex, a database available by anonymous ftp from the MIT Media Lab, and a new texture images database produced for this experiment. Twenty-nine textures were selected from Having Fun database, and thirty-five from the VisTex database. The image of each texture was divided into overlapping subimages, and the subimages were divided into two groups, a training set (from the left side of the image) and a test set (from the right half of the image).

The experiments using subsets of the feature vector show that color is indeed critical to the correct identification of these textures, even though many of the textures were predominantly shades of brown or green. The results for the combined database are shown in Figure 2 . For each subimage of every texture in

test set, we used the feature vector describing the subimage as the input to each neural net representing a texture in the database. Then we found the rank of the correct texture in each case. The graphs in Figure 2 summarize the statistics of these ranks, giving the percent that were 1st, the percent that were first or second, and so on. Our goal was to find a set of features that would prune the possible matches from the database, making it easy for the user to select from the remaining textures.

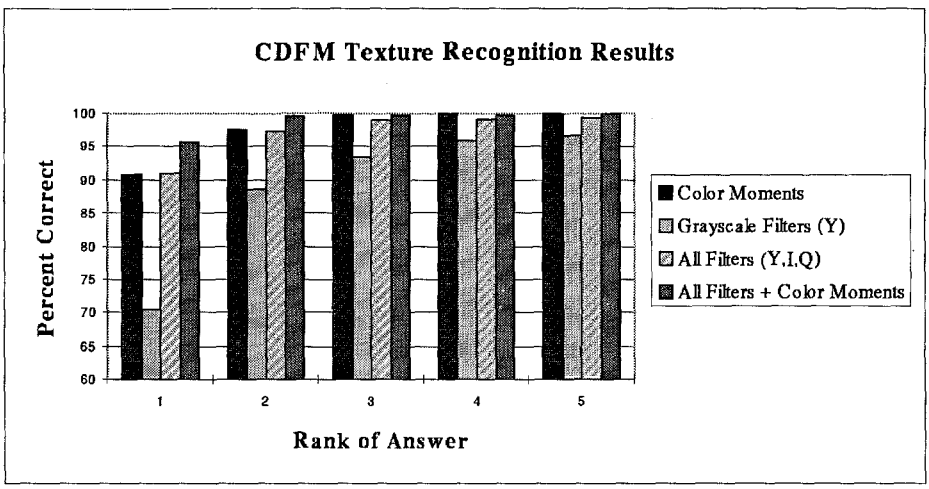


Fig. 2. Texture recognition results for the combined database.

3 Finding Objects Using Salient Features

In this section, we propose algorithms of object location and identification based on multi-resolution multi-spectral representations of salient features. To serve the purpose of representing individual objects, the most salient point on each object was chosen. Here we present a new simple algorithm (the Mean Distance (MD) algorithm) for finding the most salient point and derive other algorithms for object location and identification based on the MD algorithm. We have tested these algorithms on images with changes in orientation, lighting and scale have demonstrated that all the algorithms are quite successful.

As in the texture recognition work, we have used responses from steerable filters over 3 (YIQ), 4 linear filters and 3 scales. Instead of taking statistics of filter responses over a region, we use the responses from 36 filters centered at the same location in the image.

3.1 Salient Points and the Mean Distance (MD) Algorithm

Rao and Ballard [6] have studied similar high-dimensional feature vectors. For a given object in a particular pose, feature points were selected for the points of intersection of radial lines with concentric circles of exponentially increasing radii centered on the object centroid. The recognition rate was increased from 70% to 100% when choosing from one salient point to twenty-five salient points. They had to use many feature points in order to specify the object and reach a higher recognition rate. But some points contain better information identifying the object than others. If one picks the best of these points, one can use fewer points to identify the object. In fact, in our experiments, we picked only the most salient point to represent every object.

In this section, we will explore a simple algorithm to find *the* most salient point on the object, which algorithm is called the mean distance (MD) algorithm. The MD algorithm finds the most salient point on the object, and uses it for object location and identification. In most of our experiments, only one salient point was needed to identify the object. With more salient points, the algorithm successfully identified objects with different orientations and scales.

Over a given database of images, where we want to define the salient point, the mean distance(MD) algorithm calculates the distance from every pixel to every other pixel. Such an operation will assign every pixel a distribution of distances, from that pixel to every other pixel. The mean of the distribution, which is the mean distance from the pixel to the image, describes the property of such pixel's relationship to the whole image. If the mean distance for a point is very large, we may say this point is relatively distinct from other points. That is, in the high dimensional vector space, this vector is well separated from most of the other vectors. On the other hand, if the mean distance for a pixel is small, then it would be very similar to most of the other points in the image. That is, its representation is clustered with other similar vectors and not distinguishable.

Calculating the distance between every pair of 36-dimensional vectors in the image is quite expensive. We have studied the distributions of vector magnitudes and have found that the number of vectors decreased tremendously with increasing vector lengths. The vector with a large magnitude is more likely to be the most salient vector than a vector with a small magnitude. On the other hand, it is not necessarily true that the largest vector must be the most salient point, since the salience property is defined by the relative distances between the vectors, not the magnitudes of individuals.

The MD algorithm is based on the above hypothesis, and calculates only the distance from large vectors to the database of backgrounds to reduce the running time. It calculates the vector length distribution and sets a threshold, then only calculates the distances from those points with vector magnitudes larger than the threshold. After these calculations, it chooses the point with the largest mean distance as the salient point. The actual routines are as the following: 1. Define the area containing the object as the subimage where we want to find the salient point; 2. Calculate the vector length for every pixel in the subimage, plot the distribution of the vector lengths, and choose the threshold; 3. Calculate

distances from each pixel in the subimage with an vector length larger than the threshold chosen in step 2, record a mean distance for each pixel; 4. Find the largest mean distance and the corresponding pixel; 5. Choose this point as the salient point representing the object. Experiments have shown that the MD algorithm is not sensitive to the change of backgrounds.

Figure 3 shows the image of an apple juice bottle and its salient point with a cross on top of it, and Figure 4 graphs its corresponding salient vector.

Fig. 3. A model image of an apple juice bottle. The salient point is found by the MD algorithm, and marked with a cross on top. The frames indicate the sizes of the filters whose responses were used to represent the feature point.

3.2 Locating Objects

The location problem is defined as the problem of matching a single model object to an image at any instant. We define a simple algorithm to locate the object with a salient point. It calculates the salient point in the model image and applies it to the unknown image to find the corresponding point. The actual routines we used are as the following: 1. Given the model object, use the MD algorithm to find the best salient point and its corresponding vector; 2. Calculate the vector for every pixel in the unknown image; 3. Calculate the distance from the salient vector to the vector of every pixel in the unknown image; 4. Find the pixel position that has a minimum distance to the salient vector; 5. Such point should be the corresponding point of the salient point in the unknown image. The object is thus located to this position. In a large image database, only the responses from the salient features in each image in the database need be stored.

We have tested this algorithm with the database containing thirty six poses of a MOTT's juice bottle. Pictures of the juice bottle were taken with a 10 degree

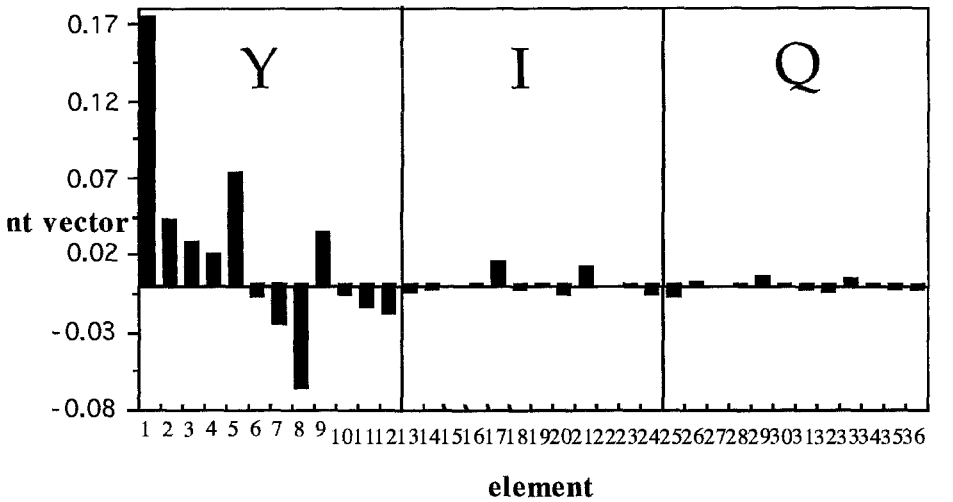


Fig. 4. Thirty-six elements of the salient vector at the point shown in the left-hand image. The three divided sections indicate the filter responses from Y, I and Q bands respectively.

difference in rotation around the vertical axis. Lighting condition and scale are unchanged in this experiment.

To test the algorithm, we picked only one salient point using the MD algorithm from a randomly picked pose image, then calculated the distance from the salient point to 36 pose images. Matching of salient point is done by setting a threshold whose value is smaller than the minimum of distances from the salient point to all points in the objects- image. Those points whose distances to the salient point are smaller than the threshold are picked as the matching points. Figure 5 shows an example of matching the salient point to 36 pose images. The matching points are marked with crosses. Seven matching images were found which cover 70 degrees in rotation. For other examples, the algorithm can always find 6-8 matching images. thus for 36 poses, we need at most 6 salient points to describe every pose of the object under rotation around the vertical axis.

4 Future Work

The aim of this work is to develop techniques suitable for finding objects and textures embedded within image and video databases. We believe that the representations employed in this work to be well suited for such domains. The steerable filter responses can be easily transformed to match rotated textures, and rotational invariants can be derived from their responses [2]. The responses

Fig. 5. An example of all the matches of the fourth salient point. The positions of the crosses indicating the points whose distances to the fourth salient vector are smaller than a threshold (0.029). The threshold is chosen smaller than the minimum (0.037) of distances from the salient point to every point in the background database.

from related scaleable filters can also be transformed to match textures at different magnifications. A number of techniques can be used to make the histogram and filter responses invariant to the intensity or to the color of the lighting, if these vary for the views of the textures within the database.

References

1. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yonker. Query by image and video content: The qbic system. *Computer*, 28:23–32, 1995.
2. H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C.H. Anderson. Overcomplete steerable pyramid filters and rotation invariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–228, 1994.
3. G. Healey and L. Wang. Illumination-invariant recognition of texture in color images. *Journal of the Optical Society of America, A*, 12:1877–1883, 1995.
4. R. Kondepudy and G. Healey. Modeling and identifying 3-d color textures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 577–582, 1993.

5. Rosalind W. Picard, Tanweer Kabir, and Fang Liu. Real-time recognition with the entire brodatz texture database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–639, 1993.
6. Rajesh P.N. Rao and Dana H. Ballard. An active vision architecture based on iconic representations. Technical Report 548, Department of Computer Science, University of Rochester, 1995.
7. Eero P. Simoncelli, William T. Freeman, Edward H. Adelson, and David J. Heeger. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory*, March 1992.
8. Markus Stricker and Markus Orengo. Similarity of color images. In *SPIE Proceedings, Vol. 2420*, 1995.
9. Michael J. Swain. Interactive indexing into image databases. In *Storage and Retrieval for Image and Video Databases*, pages 95–103, 1993.
10. Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

Graph Grammar Based Object Recognition for Image Retrieval

Christoph Klauck

FB 3 – Computer Science – AG KI, University of Bremen
Mailing address: P. O. Box 330 440, D-28334 Bremen
Telephone: ++49-421-218-4554, *FAX:* ++49-421-218-7196
E-mail: klauck@informatik.uni-bremen.de

Abstract. In order to retrieve a set of intended images from an image archive, human beings think of special contents with respect to the searched scene. The necessity of a semantics-based retrieval leads to a content-based analysis and retrieval of images. From this point of view, our project Image Retrieval for Information Systems (IRIS) develops and combines methods and techniques of computer vision and knowledge representation in a new way in order to automatically generate textual content descriptions of images. IRIS retrieves the images using a conventional text retrieval system.
This paper concentrates on the discussion of formalization knowledge for modeling concepts and on object recognition by graph grammars.
Keywords: knowledge representation, graph grammar, content-based search, image retrieval.

1 Introduction

Content-based image retrieval has been a very popular research topic for the last few years. With the on-going development of multimedia technology, the number of information systems containing image retrieval functions are increasing rapidly. Image retrieval is a key area for building large multimedia systems. Surveys and discussions of approaches and systems for image retrieval have been published by Nagy [1], Cawkell [2], and Jain [3].

The whole image retrieval process can be divided into two main steps:

1. generating the image annotations and
2. retrieving images of a collections of images.

The early idea for image annotation is the text description generated by users, e.g. the title and caption of the images as well as some additional description [4]. This approach is restricted by the effort of manual annotation and the user-dependent differences in the way of annotations. This leads very soon to inconsistencies in annotations and an undesired requirement: the person formulating the queries has to acquire knowledge concerning the criterias of the annotation generation. Furthermore, human beings cannot be asked to make the content description for thousands of images manually. Therefore, there is increasing interest in developing feature-based image retrieval systems which

integrate technologies of computer vision in extracting image features into the process of the annotation generation.

There are two kinds of research groups doing content-based image retrieval. The first one uses the well-known computer vision algorithms for generating object description and stores them in a textual form. Those annotations are then given to a retrieval system. The project IRIS belongs to this group.

The second kind of approaches is with emphasis on interactive matching similar objects based on shape, color, and texture (e.g. [5], [6]). Picard and Minka [7] use texture as a feature in their content-based retrieval method. Rather than using only one texture model, they employ several models and select the best one for discriminating a region from the rest of the image.

Until now, a major breakthrough in image retrieval has not been achieved. The following are key ideas that need to be addressed sucessfully for content-based retrieval systems to become a reality:

- definition of relevant image features and automated analysis of images,
- application of knowledge for special domains,
- development of powerful user interfaces for interactive querying.

There is no single universal approach to all the content-based retrieval problems, even in a restricted application domain. Therefore, we need domain knowledge for special applications. Chakravarthy [4] uses semantics-based relations in information retrieval. Semantics-based relations provide a way of expressing knowledge about the world. Several examples are presented to illustrate the proper incorporation of semantics-based relations into the process of matching queries to the picture representations.

The image analysis approach taken in IRIS combines color, texture, and contour for a robust description of image structures. By using domain knowledge, our approach for object recognition is able to identify various regions according to the results of image analysis. The resulting content descriptions will be stored in a textual form for automatic retrieval which is realized by the conventional system SearchManager, a boolean text retrieval system.

The next section of this paper describe the system IRIS briefly. Section 3 presents the process of knowledge-based object identification in detail. The final section gives conclusions and suggestions for future work.

2 IRIS System: Brief Overview

The IRIS system consists of two main modules (for details see [8]):

1. the image analysis module, generating the image annotations, and
2. the image retrieval module.

The image analysis module consists of four submodules: three modules extracting each one of the low level features *color*, *texture*, and *contour*. The feature extraction modules are independent of each other. Therefore, the user is able to configurate the image analysis by choosing the relevant features. The fourth module – described in the following sections – is responsible for object recognition.

Each of the low level submodules extract segments – the so-called grid elements – concerning one of the three features. These segments are described in plain ASCII text. Thereby, we receive three information sources according to these features. The sources present the low level annotations of the analyzed images. Furthermore, these annotations can be handled easily and are stored as three different aspects of the related image.

The object recognition module is based on the generated annotations of the three low level modules. First of all, the neighborhood relations in the image of the extracted segments are computed; these relations are based on the coordinates of the segments in the image. Secondly, the object recognition is realized by graph operations triggered by a graph grammar. The graph grammar presents the compiled taxonomy, which reflects the domain knowledge. The object recognition provides the information for the fourth field of the annotation belonging to the image.

In a last step of the analysis module the textual descriptions can be indexed for well-known text retrieval techniques. Text indexing and text retrieval is carried out by the conventional system SearchManager. A query is exclusively based on the text annotations and thereby the response time is minimalized.

3 Object recognition

To solve the problem of knowledge-based object recognition by the approach of syntactical pattern recognition two essential steps are necessary:

1. *Bridge* the gap between lower level (quantitative) information, i.e., the information generated from the methods described in section 2, and the atomic entities of the higher level (qualitative) information, i.e., the primitive objects described in a thesaurus. The result of this first step are hypothesies concerning the primitive objects.
2. *Combine* the primitive objects according to the compositional semantics of more complex objects described in a thesaurus – our knowledge base. A hypothesis used in one description of the analyzed image becomes a thesis.

Inherent in the information about color, texture and contour of the image analysis phase is the information about the topological relations in the image data between these different segments as illustrated in Fig. 1. These *neighborhood* relations are distinguished by three cases: *overlaps, meets, contains* and their inverse relations.

One fundamental assumption of the IRIS system is, that these neighborhood relations restrict the recognition of objects, i.e., an (primitive) object is built upon segments which are in this neighborhood relation. By this assumption the process of object recognition can be treated as the process of graph transformation, i.e., the process of graph rewriting.

In the IRIS system a graph parser – the so-called *GraPaKL* [9] – is used to solve the object recognition phase. The underlying graph grammar formalism and the parser algorithm are described later on. Within the IRIS system two graph grammars are used:

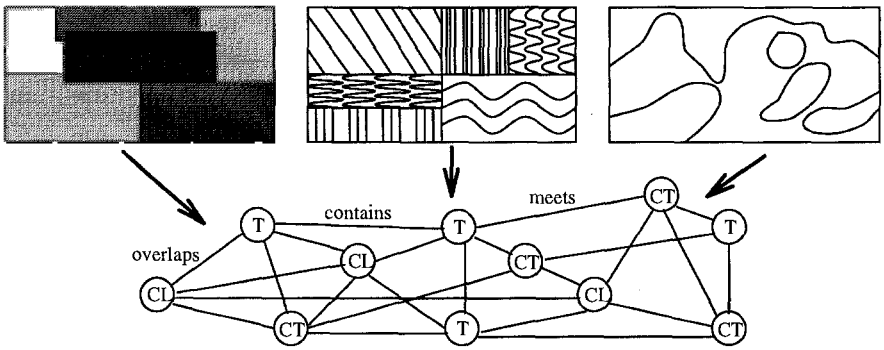


Fig. 1. Come together: Color (CL), Texture (T) and Contour (CT).

1. a grammar to bridge the gap between lower level information and primitive objects, and
2. a grammar to combine the primitive objects.

An example grammar is given in Fig. 3.

The grammars are compiled out of our knowledge base, represented in a thesaurus. In this sense the model of our recognizable world is represented within this thesaurus.

3.1 1-NRCF Graph Grammars and GraPaKL

In the following, we define the terminology of neighborhood-controlled node labeled and node attributed feature graph grammars (1-NRCFGG) as specified in [9]. Surveys and detailed introduction to graph grammars can be found e.g. in [10, 11].

In our paper the term *feature graph* represents a finite undirected node attributed and node labeled graph, in the sequel shortly called *graph*. The interpretation of the edges in our graph's is restricted to a kind of topological neighborhoodness (*nr*), i.e., a kind of *nr*-inheritance is guaranteed.

A *neighborhood-controlled node labeled and node attributed feature graph grammar* (1-NRCFGG) is a grammar on our graph's, where the productions left hand side (lhs) is described by one node and the productions right hand side (rhs) is given by any graph (see Fig. 3). Additional conditions are associated with each production. These conditions are often called *dependency relations* and serve the purpose of proving, passing or generating information by calculating attributes. The global *embedding specification* is given by the *nr*-inheritance.

Our chart parser – called *Graph Parser Kaiserslautern* (GraPaKL) – is described in detail in [9]. The algorithm is an extension to the one introduced in [12]. The approaches differ mainly in the use of a heuristic-guided search, the ability to specify overlappings, the distinction of nodes by their components and the use of labels structured by a subsumption-hierarchy.

The main algorithm of our chart parser – see Fig. 2 – consists of three rules: initialize, choose, and combine. These rules operate on the agenda and the chart which are sets of complete (cp) and partial (pp) instances of productions – the so-called *patches*. A complete patch is one where every node on the rhs is instantiated. Furthermore, a partial patch is one where some nodes, i.e., not all nodes on the rhs are instantiated. The chart contains all patches which are not contained in the agenda. Its purpose is, to quickly retrieve a small, but complete superset of all patches, which are combinable with a given patch. The superfluous patches can then be eliminated by evaluating the conditions.

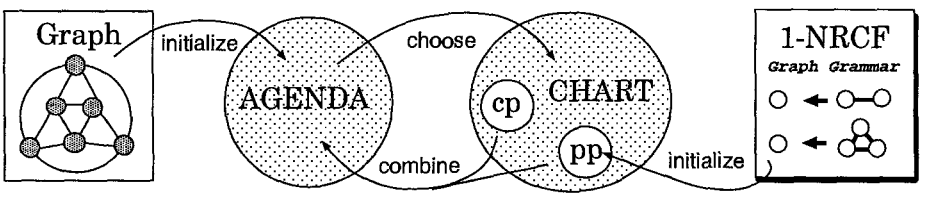


Fig. 2. The main algorithm of GraPaKL.

The agenda is initialized with the (starting) graph's nodes (regarding as rule with empty rhs) and the chart is initialized with each production (where no node of the rhs is instantiated). choose removes any patch from the agenda and moves it into the chart. combine merges a chosen cp with all pp in the chart – respectively a chosen pp with all cp in the chart. If a merge was successful, a new pp (or probably cp) for every successful combination will be created. choose and combine are applied alternately until the agenda is empty. In this case the chart contains all possible parses, i.e., all recognizable objects.

The ability to avoid the combinatoric explosion inherent to the graph parsing problem includes two kinds of heuristics: the incorporation of a heuristic-guided search is begun by ordering the production's nodes, i.e., to find the restrictive part of a productions rhs first, and then by ordering the production's themselves (in our case realized with a neural network [13]). Additionally, the use of labels helps to reduce the parser's search space. The labels are given by the names of the objects.

3.2 Knowledge Representation

All knowledge needed for object recognition has to be stored in our knowledge base. This tool is an extension and combination of a logic-based thesaurus [14], a KL-ONE (Knowledge Language ONE) like system [15] and a front-end visualization tool [16]. The main components of this knowledge base – representing the model of our object world – are the visualization, representation and consistency checks components.

The *visualization* component is the graphical user interface of the thesaurus mainly based on *daVinci*, the front-end visualization tool [16]. It offers the user an easy possibility to enter, view and manipulate definitions of objects (see Fig. 4). In Fig. 3 the visualization of a typical object is shown.

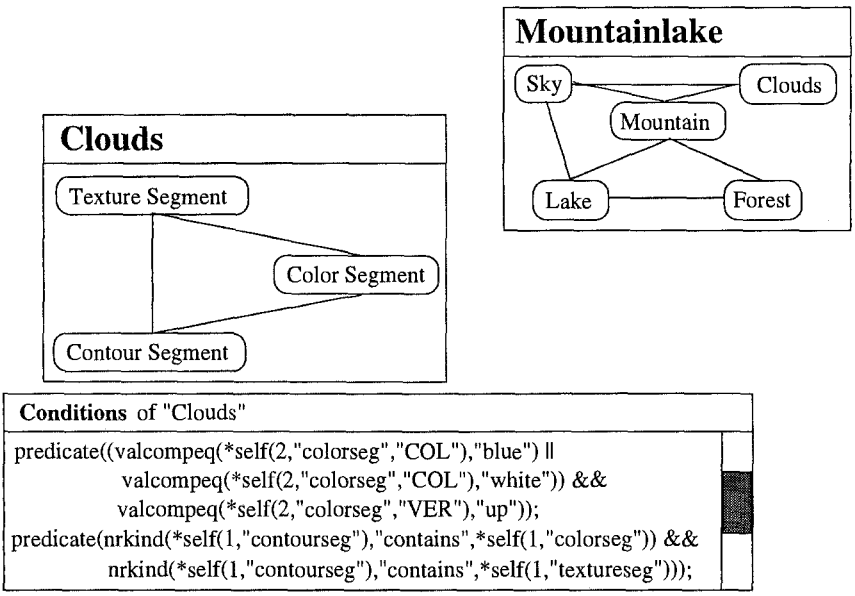


Fig. 3. Object `Clouds` and `Mountainscene`.

The *representation* component stores the entire knowledge. Several functions are provided for access and modification of the knowledge base. Integrated in this component is a concept language based on KL-ONE [15] and a logic-based thesaurus [14].

Our tool offers several *consistency checks* and verifies the defined knowledge for soundness and completness. This will be performed during the knowledge acquisition/editing phase. The tests are adapted to our purpose, the aim is to prevent the description of objects. This offers the user the possibility to detect and to eliminate most of the errors as early as possible.

Some checks are performed when new or changed knowledge is saved by the user to the knowledge base. The complete check is only performed on a request from the user.

3.3 Strategies for modeling the domain knowledge

Following the approach of syntactical pattern recognition, a graph grammar is a powerful method to handle object recognition by substitution of topological

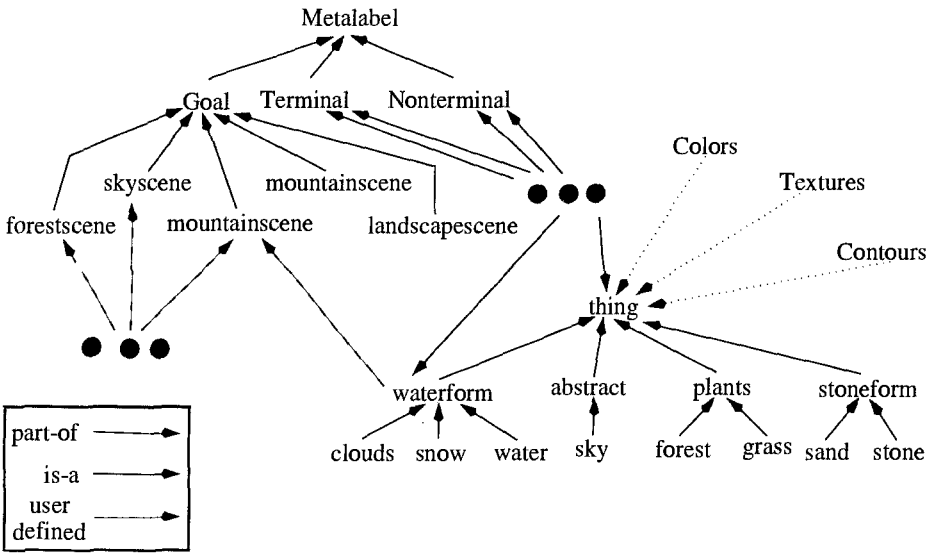


Fig. 4. Object Definitions.

graphs. The trick consists in finding an adequate and consistent model of the domain. This paragraph concentrates on the underlying modeling-strategies of one of the current IRIS grammars, which treats the landscape domain. A larger grammar treating the domain of technical drawings contains 105 rules and is described in [17].

The graph grammar consists of three different object types: goal, terminal and nonterminal (see Fig. 4). Terminal nodes are represented by the input of the color, texture and contour module. Therefore, the nonterminal nodes are composed of color, texture and contour segments. Hence it follows that the nonterminal nodes are divided into different object-classes: the primitive objects, which are just supported by the color, texture and contour segments (specifying the grammar about the primitive objects) and the complex objects, which rest on the primitive objects.

The strategie on modeling the primitive objects is that they consist always of a color, texture and contour segment (see Fig. 3). The size of the color and the texture segments are only allowed to differ by two and both segments are contained by the contour segment. Thereby, the correspondence of the color- and the texture-annotation to the same region should be given. In the current grammar we modeled eight primitive objects each defined by one grammar rule: `sky`, `clouds`, `forest`, `grass`, `sand`, `snow`, `stone`, and `water`.

The current strategies on modeling complex objects incorporate:

- Complexed objects are composed of primitive objets. To reduce the number of rules for a definition of a concept supersort relations are introduced.

- The primitive objects are generally specified by their size, e.g. the complex object should be dominated by the forest segment.
- The primitive objects are related by the topological relations `meets`, `contains` and `overlaps` to insure their neighborhood.

Up to now, five complex objects have been modeled. The number of rules is notated in parentheses: `landsscapescene` (4 rules), `mountainscene` (4 rules), `mountainlakescene` (8 rules), `skyscene` (6 rules) and `forestscene` (6 rules).

Our current grammar is just composed of three layers: Segments as terminals (first level), primitive objects as nonterminals (second level) and complex object as goals (third level).

4 Conclusions and Future Work

The described results demonstrate how the techniques in computer vision, graph grammar, and text retrieval can be combined in a rather new way to produce an efficient image retrieval system. On the one hand, IRIS offers image retrieval by using low level features. On the other hand, the system presents semantics-based image retrieval by using natural language concepts.

The system is implemented on IBM RS/6000 using AIX and has already been tested with an archive comprising 1218 images. Concerning this database 615 images are relevant to the landscape grammar. Precision and recall do not variate significantly from one complex object to another.

Enlarging the image database with new images was most important. The different values of recall and precision have the following explanation: In general, the recall was reduced by a maximum of 0.05 by the enlargement of the image database. The precision of the concept sky scene was increased but nevertheless resulted in 0.67. The precision of the concepts mountain scene and forest scene were constant and the precision of landscape scene improved to 0.62. These results demonstrate that the IRIS system is capable of recognizing the semantics of images adequately concerning the concepts landscape scene, sky scene, mountain scene and forest scene.

Until now, IRIS has been based on boolean retrieval strategies. But in general, different images belong to simple and complex objects with different probabilities. Therefore, probabilistical retrieval techniques will be tested to enhance and rank the recall. Additionally IRIS will be applied to differend applications to become a more broader and reliable framework for image retrieval.

5 Acknowledgments

This research was performed during the project IRIS at the Artificial Intelligence group of the University of Bremen. It was carried out in cooperation with the German Software Development Laboratory (GSDL) of IBM. Special thanks to Th. Hermes, Prof. O. Herzog, J. Kreyß and J. Zhang.

References

1. G. Nagy, "Image databases", *Image and Vision Computing*, vol. 3, no. 3, pp. 111–117, 1985.
2. A.E. Cawkell, "Imaging systems and picture collection management: a review", *Information Services & Use*, vol. 12, pp. 301–325, 1992.
3. R.C. Jain, Ed., *NSF Workshop on Visual Information Management Systems*, Workshop Report, Computer Science and Engineering Division, The University of Michigan, Ann Arbor, Mich., 1992.
4. A.S. Chakravarthy, "Toward semantic retrieval of pictures and video", in *Proc. Riao'94, Intelligent Multimedia Information Retrieval Systems and Management.* New York, 1994, pp. 676–686.
5. J. Brolio, B.A. Draper, J.R. Beveridge, and A.R. Hanson, "SR: a database for symbolic processing in computer vision", *Computer*, vol. 22, no. 12, pp. 22–30, 1989.
6. A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for content-based manipulation of image database", in *Proc. of SPIE on Storage and Retrieval for Image and Video Databases.* San Jose, Calif., 1994, pp. 34–47.
7. R.W. Picard and T.P. Minka, "Vision texture for annotation", *Multimedia Systems*, vol. 3, no. 1, pp. 3–14, 1995.
8. Th. Hermes, Ch. Klauck, J. Kreyß, and J. Zhang, "Image retrieval for information systems", in *Proc. of SPIE - The Inter. Soc. for Optical Engineering, Storage and Retrieval for Image and Video Databases*, 1995.
9. Ch. Klauck, *Eine Graphgrammatik zur Repräsentation und Erkennung von Features in CAD/CAM*, vol. No. 66 of *DISKI*, infix-Verlag, St. Augustin, 1994, Dissertation (Ph.D. Thesis), University of Kaiserslautern.
10. H.-J. Kreowski and G. Rozenberg, "On Structured Graph Grammars. I", *Information Sciences*, vol. 52, pp. 185–210, 1990.
11. H.-J. Kreowski and G. Rozenberg, "On Structured Graph Grammars. II", *Information Sciences*, vol. 52, pp. 221–246, 1990.
12. R. Lutz, "Chart Parsing of Flowgraphs", in *Proceedings of the 11th International Joint Conference on AI (IJCAI).* 1989, pp. 116–121, Morgan Kaufmann.
13. Ch. Klauck, "Heuristic Driven Chart-Parsing", in *Proceedings of the 5th International Workshop on Graph Grammars and their Applications to Computer Science 94*, 1994, pp. 107–113.
14. S. Goeser, "A Logic-based Approach to Thesaurus Modelling.", in *Proceedings of the International Conference on Intelligent Multimedia Information Retrieval Systems and Management (RIAO) 94.* 1994, pp. 185–196, C.I.D.-C.A.S.I.S.
15. P. Hanschke, A. Abecker, and D. Drollinger, "TAXON: A Concept Language with Concrete Domains.", in *Proceedings of the International Conference on Processing Declarative Knowledge (PDK) 91.* 1991, pp. 411–413, Springer-Verlag, LNAI 567.
16. M. Fröhlich and M. Werner, "Demonstration of the interactive Graph Visualization System daVinci.", in *Proceedings of DIMACS Workshop on Graph Drawing '94.* 1994, pp. 266–269, Springer-Verlag, LNCS 894.
17. Ch. Klauck and J. Mauss, "Feature Recognition in CIM", *Integrated Computer-Aided Engineerin: Special Issue on AI in Manufacturing and Robotics*, vol. 1, no. 5, pp. 359–373, 1994.

Similarity Matching

Simone Santini and Ramesh Jain

Visual Computing Laboratory
University of California, San Diego*

Abstract. Image databases will force us to rethink many of the concepts that led us so far. One of these is *matching*. We argue that the fundamental operation in a content-indexed image database should not be matching the query against the images in the database in search of a "target" image that best matches the query. The basic operation in query-by-content will be ranking portions of the database with respect to similarity with the query. What kind of similarity measure should be used is a problem we begin exploring in this paper. We let psychological experiments guide us in the quest for a good similarity measure, and devise a measure derived from a set-theoretic measure proposed in the psychological literature, modified by the introduction of fuzzy logic.
We report one experiment comparing this measure with other proposed in experimental psychology.

1 Introduction

What makes a multimedia database different from the databases we have been using until now? Many things, one might say: there are different ways to express a query (a sketch, an image...), different devices used for input/output, a strong emphasis on user interface, a difference of several orders of magnitude in the amount of data we might expect to deal with, and an increasing necessity to distribute databases efficiently in vast geographic areas.

All these differences will, and do, play an important rôle in designing the architecture and the solutions for the design of an images database, but there is also a more fundamental difference we must consider: image databases must be designed to handle a different type of search than conventional databases. While a search in a traditional database always results in a clear distinction between the elements that match the query and the elements that don't, this is no longer true when we search images by content. Although it is certainly possible to match one image against the other looking for equality, the most natural approach to image databases is based on *similarity*.

When we make a search in a traditional database, we always have a clear target in mind: we look for those records that satisfy a certain query, and only those records. When making a search in an image database, most of the time, we don't have any particular target in mind: we might ask for a photograph with a certain color tone, or with certain

* Send correspondence to: Simone Santini, Department of Computer Science and Engineering, UCSD, 9500 Gilman Drive, La Jolla, CA 92093-0114; email: ssantini@cs.ucsd.edu, or Ramesh Jain, Department of Electrical and Computer Engineering, UCSD 9500 Gilman Drive, La Jolla, CA 92093; email: jain@ece.ucsd.edu

elements placed in a given way. There might be no image in the database that "exactly" matches the query, and, if there is, we might not even be interested in that (think about specifying the query as a sketch: you don't want the database to return a sketch like the one you just did). Exactly matching the query doesn't make much sense in an image database.

The basic operation in an image database is the *ordering* of a portion of the database with respect to the similarity with the query. Just like the matching is the single most important operation in traditional databases—since it decides, ultimately, which data satisfy the query—so similarity measurement is the single most important operation in image databases. Unlike matching, though, there is not an obvious definition of similarity.

Since the results of the query must be judged by a human user, it seems natural to let the *human* concept of similarity drive us in the definition of a similarity measures. Psychologists have been experimenting for some 70 years trying to define the properties of human similarity judgment. Many of these models succeed in explaining qualitatively some experimental findings, but few of them are in a mathematical form suitable for application to automatic computation.

In section 2, we briefly review some of the most debated models of human similarity. The emphasis of this section will be on whether human similarity can or cannot be modeled as a distance in a metric space. We will review some of the models presented in the psychological literature, and the assumptions they make. Our goal is to present some general ideas and to provide pointers to the psychological literature, to which the interested reader is referred.

In section 3, we analyze a little more closely one of the most successful psychological models and, in section 4, we discuss its *fuzzyfication*, that turns out to be necessary if the similarity measure has to be used in all but the most simple laboratory experiments. Finally, in section 5, we present some of the result we obtained using these metrics, when applied to the determination of the similarity between human faces.

2 Metric or not Metric?

In the past decades, a number of psychologists developed experiments to measure the human similarity perception, and devised models to explain the results they obtained. There have been a number of different models, stemming from very different assumptions, and trying to explain the outcome of different experiments. These models differ in several features, but one of the most characteristic differences—and one of the most debated points in the psychological community—is whether human similarity measurement follows the metric axioms and, if not all, which axioms are satisfied and which are not.

Suppose S_A and S_B are two stimuli, represented as vectors in some space of suitable dimension, and let the similarity between the two be measures via a *psychological distance* function $d(S_A, S_B)$. In general, the assumption is made that the *perceived similarity* d is different from the *judged similarity* δ, and that the two are related by a monotonically nondecreasing function g:

$$\delta(S_A, S_B) = g[d(S_A, S_B)] \tag{1}$$

Given that experiments only give access to δ, the judged similarity, it is not clear whether there is any scientific basis to postulate the existence of the function d. While researchers try to understand the properties of d, they only can determine those of δ. Because of this, some properties that can be postulated for d cannot be verified experimentally, since they don't necessarily translate to δ.

If d is a metric function, it has the following characteristics:

Constance of self-similarity: for all stimuli S it is $d(S, S) = 0$. This is an experimentally testable property, since it implies $\delta(S_A, S_A) = \delta(S_B, S_B)$ for all stimuli S_A, S_B.

Minimality: for all stimuli S_A and S_B, it is $d(S_A, S_A) \leq d(S_A, S_B)$. This is also an experimentally testable property, since it implies $\delta(S_A, S_A) \leq \delta(S_A, S_B)$, due to the monotonicity of the function g.

Symmetry: for all stimuli S_A and S_B, it is $d(S_A, S_B) = d(S_B, S_A)$. This property too can be tested experimentally.

Triangle inequality: for all S_A, S_B, S_C, it is

$$d(S_A, S_C) \leq d(S_A, S_B) + d(S_B, S_C)$$

This property cannot be tested experimentally. Even if d satisfies the triangle inequality, δ might not, or vice versa.

The constance of similarity has been refused by Krumhansl [6], that proposed a modification to the distance model in order to allow for its violation: in Krumhansl's model, the distance between stimli S_A and S_B is to be expressed as

$$\delta(S_A, S_B) = h(S_A) + d(S_A, S_B) \tag{2}$$

where d is a metric distance, and h is a function that depends on the density of the stimuli around the stimulus S_A. This model also account for violation of symmetry, which is generally acknowledged not to hold for human similatity assessment, as proved experimentally by a number of researchers [12, 7, 8].

Monotonicity seems to hold, although in [12] it is argued that

The triangular inequality is the most debated and troublesome of the metric axioms, since, according to the model, the satisfaction or violation of this property by the function d is not accessible to experimentation. It is common wisdom, however, that the triangular inequality does not hold for human similarity perception [13, 1].

Tversky and Gati [13] proposed a substitute for the triangle inequality that they call *corner inequality*. If x_1y_1, x_2y_2, and x_3y_3 are three points in a two-dimensional space, and $x_1 < x_2 < x_3$ and $y_1 < y_2 < y_3$, the corner inequality holds if[2]

$$d(x_1y_1, x_3y_1) > d(x_1y_1, x_2y_2) \text{ and } d(x_3y_1, x_3y_3) > d(x_2y_2, x_3y_3) \tag{3}$$

or

$$d(x_1y_1, x_3y_1) > d(x_2y_2, x_3y_3) \text{ and } d(x_3y_1, x_3y_3) > d(x_1y_1, x_2y_2) \tag{4}$$

[2] These conditions are slightly less general than those originally postulated by Tversky and Gati. The reader is referred to [13] for the more general conditions.

It can be proved that, if the function g relating d and δ in (1) is monotonic then the corner inequality holds for δ if and only if it holds for d that is, unlike the triangular inequality, compliance with the corner inequality can be verified experimentally and, indeed, has been verified *not* to hold for human observers.

To sum up, a model of human similarity should ideally present the following characteristics:

1. Non constant self-similarity
2. Asymmetry
3. Violation of the corner inequality

2.1 Some Geometric Models

In spite of evidence that constancy of self-similarity and symmetry do not hold in general, metric models of similarity are quite popular, mainly because of their simplicity and the powerful mathematical apparatus available to study them. Also, it must be considered that a property that is proved not to hold in an experiment with given stimuli can hold for a different type of stimuli. For instance, similarity of certain global properties, like overall color, seems to be metric.

In all this section, we will consider the problem of determining the perceptual similarity or the perceptual distance between two objects S_A and S_B, being described by the sets of features $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$, respectively. The feature vectors are two elements of $\mathbb{R}^n$.

Euclidean Distance The simplest way to compute the perceptual distance between the stimuli is by the Euclidean distance between the two vectors:

$$d_e(S_a, S_B) = \left[\sum_{i=1}^{n}(a_i - b_i)^2\right]^{\frac{1}{2}} \tag{5}$$

Although there is quite conclusive evidence that the perceptual distance in human is not Euclidean, this distance is still used as a reference against which models are compared.

City Block Distance In a series of experiments with rectangles changing shape and size, Attneave [2] found a good agreement between the experimental data and the *city block* model:

$$d_c(S_a, S_B) = \sum_{i=1}^{n}|a_i - b_i| \tag{6}$$

Thurstone-Shepard Models In these models, [11, 9, 10, 5, 4], it is postulated that similarity is based on a *momentary distance*, which is assumed to be a Minkowski distance:

$$d = \left[\sum_{k=1}^{n} |x_k - y_k|^{\gamma} \right]^{\frac{1}{\gamma}}, \quad \gamma \geq 1. \tag{7}$$

The similarity between two stimuli is a function g of the distance, usually assumed to be of the form:

$$g(d) = \exp(-d^{\alpha}) \tag{8}$$

where α is a positive parameter. The most common models derived from this family are the Euclidean/Gaussian model (for $\gamma = \alpha = 2$) and the City-block/Exponential model (for $\gamma = \alpha = 1$).

3 The Feature Contrast Model

The failure to satisfy the requirements outlined in sec. 2 with metric models has led to the explorations of alternative ways of similarity assessment. One of the most influential papers in the field, published by Tversky [12], uses a set-theoretic model (see also [3]).

Let a, b be two stimuli, described by the presence or absence of features drawn from a common feature set. Let A and B be the set of features present in the stimuli a and b, respectively. Also, let $s(a, b)$ be a measure of the similarity between a and b. Tversky considers similarity functions of the form

$$s(a, b) = F(A \cap B, A - B, B - A) \tag{9}$$

and then proves that, under reasonable hypotheses, the following *representation theorem* holds:

Theorem 1. *Let s be a similarity function of the form (9). Then (under reasonable hypotheses, see [12]) there are a similarity function S and a non-negative function f such that, for all a, b, c, d:*

$$S(a, b) \geq S(c, d) \Longleftrightarrow s(a, b) \geq s(c, d) \tag{10}$$

$$S(a, b) = f(A \cap B) - \alpha f(A - B) - \beta f(B - A) \tag{11}$$

This result implies that any similarity ordering can be obtained using a linear combination (contrast) of a function of the common features ($A \cap B$) and of the distinctive features ($A - B$ and $B - A$). This representation is called the *contrast model.*

The feature contrast model has proved very successful in explaining the outcome of experiments; in particular, it explains asymmetry, since

$$S(a, b) > S(b, a) \text{ whenever } f(A) > f(B) \tag{12}$$

The models accounts for variation of self-similarity ($S(a, a) = f(A)$) and violation of the corner inequality.

4 The Fuzzy Feature Contrast Model

The feature contrast model has been quite successful in explaining the characteristics of human similarity assessment in laboratory experiments, where the subjects had to rate simple stimuli. It encounter problems, however, when we try to apply it to the typical situations we have in computer vision. The main problem of the approach is that it requires the stimuli to be defined by a set of binary (yes/no) features. This can easily be accommodated in the controlled world of a psychology laboratory, but analysis of real world images is more likely based on a series of continuous measurements, grouped in a feature vector.

For instance, in a faces database, we may have feature vectors made of geometric measurements of a face, such as the width of the mouth or the length of the nose, while the feature contrast model would require binary features such as the presence of a wide mouth or of a long nose.

In this section, we modify the feature contrast model into the *fuzzy feature contrast model.* Consider again the problem of measuring the similarity between faces. A face is characterized by a number of features of different types but, for the following discussion, we will only consider *geometrical features*, since these lead naturally to predicate features. It seems pretty intuitive that face similarity is influenced by things like the size of the mouth, the shape of the chin, and so on. Also two faces with big mouths will be, all other things being equal, more similar than two faces one with a big mouth and one with a small mouth.

A predicate like *the mouth of this person is wide* can be modeled as a fuzzy predicate whose truth, in the first approximation, is supposed to be based only on the measurement of the width of the mouth. This is only an approximation, since, for instance, mouths of the same width will appear different depending on the measures and sistribution of other measures in the face. Once we have measured the width of the mouth, we can apply two truth functions like those in Fig. 1 to determine the truth of the predicates *the mouth is wide* and *the mouth is narrow.*

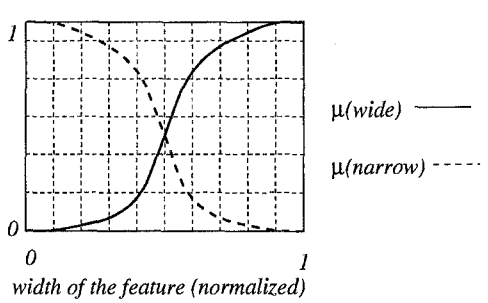


Fig. 1. Typical membership functions for narrowness and "wideness"

In general, we have an image I and a number of measurements ϕ_i on the image.

We want to use these measurements to determine the truth of *n fuzzy predicates*. Some care must be taken to define the truth value of a fuzzy predicate. We use the following definition:

Definition 2. Let Ω be a universe of discourse, and $\phi_i : \Omega \rightarrow \mathbb{R}^+$ a set of measures on the elements of Ω. Let P be a predicate on Ω, and let M_P be the (fuzzy) set of the elements $\omega \in \Omega$ for which the predicate is true. Let $\mu_M = \mu_M(\phi)$ be the membership function of M_P. Then the *truth value* of the predicate $P\omega$ is

$$\mu(P\omega) = \mu_M(x(\omega))$$

For instance, the truth value of the predicate "The mouth of X is wide" equals the membership of X into the set of persons with a wide mouth.

From the measurements ϕ_i we derive the truth values of a number p of fuzzy predicates, and collect them into a vector:

$$\mu(\phi) = \{\mu_1(\phi), \ldots \mu_p(\phi)\} \tag{13}$$

We call $\mu(\phi)$ the (fuzzy) set of *true predicates* on the measurements ϕ. We use this fuzzy set as a basis to apply Tversky's theory.

In order to apply the feature contrast model to the fuzzy sets $\mu(\phi)$ and $\mu(\psi)$ of the predicates true for the measurements ϕ and ψ, we need to do three things:

- Compute the fuzzy set $\mu(\phi) \cap \mu(\psi)$
- Compute the fuzzy set $\mu(\phi) - \mu(\psi)$ (and, by the same definition, $\mu(\psi) - \mu(\phi)$)
- Choose a suitable salience function f.

For the *saliency function* f in eq. (11), we take the cardinality of the fuzzy set $\mu = \{\mu_1 \ldots \mu_p\}$:

$$f(\mu) = \sum_{i=1}^{p} \mu_i \tag{14}$$

The intersection of the sets $\mu(\phi)$ and $\mu(\psi)$ is defined in the traditional way:

$$\mu_\cap(\phi, \psi) = \{\min\{\mu_1(\phi), \mu_1(\psi)\}, \ldots \min\{\mu_p(\phi), \mu_p(\psi)\}, \} \tag{15}$$

For the difference of two fuzzy sets, we use the definition:

$$\mu_-(\phi, \psi) = \{\max\{\mu_1(\phi) - \mu_1(\psi), 0\}, \ldots \max\{\mu_p(\phi) - \mu_p(\psi), 0\}, \} \tag{16}$$

which has the advantage that $A - A = \emptyset$. With these definitions, we can write the Tversky's similarity function between two fuzzy sets $\mu(\phi)$ and $\mu(\psi)$ corresponding to measurements made on two images as:

$$\begin{aligned} S(\phi, \psi) = & \sum_{i=1}^{p} \min\{\mu_i(\phi), \mu_i(\psi)\} \\ & - \alpha \sum_{i=1}^{p} \max\{\mu_i(\phi) - \mu_i(\psi), 0\} \\ & - \beta \sum_{i=1}^{p} \max\{\mu_i(\psi) - \mu_i(\phi), 0\} \end{aligned} \tag{17}$$

We refer to the model defined by eq. (17) as the *Fuzzy Features Contrast* (FFC) model.

5 Similarity of Faces

In this section we present an experimental comparison of the results obtained by some of the similarity measures discussed in the previous sections. The testbed we use is that of the similarity of human faces. Similarity of faces is a complex issue, that depend on a number of factors, like the color and the shape of the hair, the texture of the skin, the geometry of the face components, and so on. In this experiment, we have chosen a limited approach, and we will measure similarity based only on geometric measures.

Geometrical Measurements and predicates The geometric measurements we derive from a face image are described in Fig. 2. Opportune combinations of these measure-

Fig. 2. These 6 measures are taken from a face image to provide support to the fuzzy predicates used for the similarity assessment.

ments provide support for the 8 predicates of Tab. 1. These predicates can be collected in a set of features, and used to compute Tversky similarity. The FFC similarity model uses the truth value of the predicates, while metric distances are based on the geometric measurements.

Experimental results For the experiments, we use ten faces taken from a subset of the MIT face images database. All the images in this subset are mug shots of faces of approximately the same scale and with approximately constant illumination.

A typical stimulus-response result is presented in Fig. 3. We note several differences in the responses given by the similarity measures we considered. For instance, the person that was deemed "more similar" by three of the metric models (and was deemed "second similar" by the fourth) is placed only in fifth position by the FFC measure.

Predicate	Supporting quantity	Membership type
Long face	b	"X" is large
Long chin	d	"X" is large
Wide mouth	e	"X" is large
Round face	g/b	"X" is a 1/1 ratio
Long nose	a	"X" is large
Square face	$(g-f)/g$	"X" is small
Large face	g/b	"X" is large
Large chin	f/b	"X" is large

Table 1. Predicates used for similarity evaluation, and measured quantities that support their truth. All these measures are normalized with respect to the distance between the eyes. The column "Membership type" contains the type of membership function used to assess the true of that predicate based on the supporting measure.

The only difference between the two instances of FFC is that the face that is ranked "most similar" when $\alpha > \beta$ is ranked third when $\beta > \alpha$. Since the parameters for the case $\beta > \alpha$ were chosen with the declared purpose of finding the maximum difference, this indicates that, in this case, FFC is quite insensitive to the values of α and β. This makes the FFC model "almost" symmetric in the case of faces. We don't know of any experimental data that might confirm nor disprove this symmetry in the judgment of face similarity for human observers.

The only difference between the Attneave an Euclid distances is the inversion of the faces in the third and fifth position. The Shepard model is much less consistent: comparing the case $\gamma = 6$ with the case $\gamma = 1$we see that many faces are ranked differently. The average displacement, from the two cases is 2, versus an average displacement of $1/2$ between the two instances of FFC.

References

1. F. Gregory Ashby and Nancy A. Perrin. Toward a unified theory os similarity and recognition. *Psychological Review*, 95(1):124–150, 1988.
2. Fred Attneave. Dimensions of similarity. *Americal Journal of Psychology*, 63:516–556, 1950.
3. Michael M. Cohen and Dominic W. Massaro. On the similarity of categorization models. In Gregory F. Ashby, editor, *Multidimensional Models of Perception and Cognition*, Scientific psychology series. Lawrence Erlbaum Associates, Inc, 1992.
4. Daniel M. Ennis and Normal L. Johnson. Thurstone-shepard similarity models as special cases of moment generating functions. *Journal of Mathematical Psychology*, 37:104–110, 1993.
5. Daniel M. Ennis, Joseph J. Palen, and Kenneth Mullen. A multidimensional stochastic theory of similarity. *Journal of Mathematical Psychology*, 32:449–465, 1988.
6. Carol L Krumhansl. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, 85:445–463., 1978.

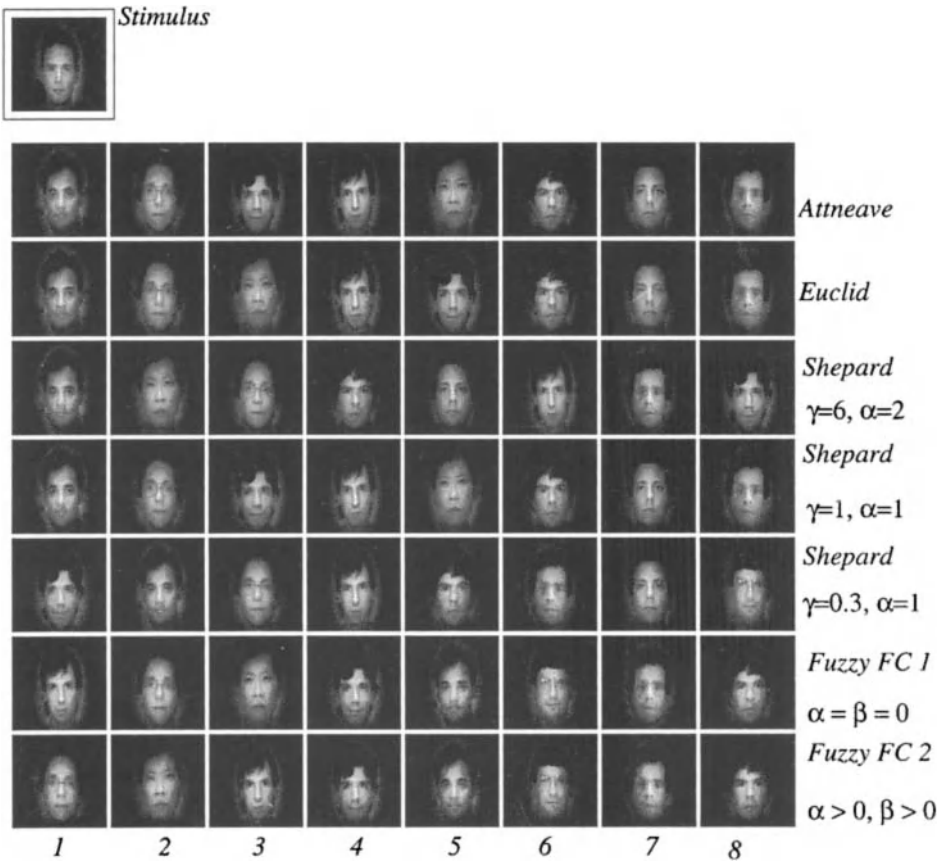


Fig. 3.

7. Eleanor Rosh. Cognitive reference points. *Cognitive Psychology*, 7:532–547, 1975.
8. E. Z. Rothkopf. A measure of stimulus similarity and errors in some paired-associate learning tasks. *Journal of Experimental Psychology*, 53:94–101, 1957.
9. Roger N. Shepard. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22:325–245, 1957.
10. Roger N. Shepard. Toward a universal law of generalization for physical science. *Science*, 237:1317–1323, 1987.
11. L. L. Thurstone. A law of comparative judgement. *Psychological Review*, 34:273–286, 1927.
12. Amos Tversky. Features of similarity. *Psychological review*, 84(4):327–352, July 1977.
13. Amos Tversky and Itamar Gati. Similarity, separability, and the triangle inequality. *Psychological Review*, 89:123–154, 1982.

Document Image Analysis

A Document Image Analysis and Recognition System for Japanese Family Registration

Tomio AMANO, Kazuharu TOYOKAWA,
Takashi MANO, and Shuhji TORIYAMA

Tokyo Research Laboratory, IBM Japan Limited
1623-14, Shimotsuruma Yamato-shi, Kanagawa-ken 242 Japan
TEL: 81-462-73-4605 FAX: 81-462-73-7413
e-mail: amano@trl.ibm.co.jp

Abstract. A family registration data entry system with functions for automatic form layout analysis and character recognition was developed. The layout analysis module first detects characters and ruled lines by using information on the top and bottom boundaries of smeared black components. It then determines the layout and identifies each field in the layout by comparing predefined models with detected lines. Character strings in the fields are recognized and matched with a dictionary to check whether a sequence is plausible as a Japanese word or not. The text data are registered in a database after they have been examined by an operator and keywords have been extracted. This system was actually used for the initial entry of typed family registration forms in Tokyo's Toshima Ward, which contributed to establish the first computerized family registration system in Japan.

1 Introduction

This paper describes a system for entering family registration data into a computer database, with functions for automatic form layout analysis and character recognition. Running since March 1995, it is one of the first computer-based registration systems in Japan, and facilitated the entry into a database of about 137,000 family registers in Tokyo's Toshima Ward.

A family register in Japan is a formal record of family events such as births, marriages, adoptions, and deaths. It contains records of a Japanese citizen's family members, parents, and children. The Ministry of Justice (MOJ) is responsible for the administration of family registers, but the day-to-day running of tasks such as updating records every time a family event is reported and certifying family registration in response to individual requests is delegated to local city, ward, town, and village authorities. Until recently, the family registration law stipulated that hard-copy records must be used for these purposes. As a result, no computer-based filing and retrieval systems were used for family registration.

We have been involved since 1989 in a formal study group entrusted by the MOJ with the task of investigating ways of improving the family registration

system. After the study group had proposed a new computer-based family registration system, the MOJ submitted to the Diet in May 1994 a bill calling for a new family registration law. The bill was passed in July 1994 and ratified in December 1994. Although a computerized family registration system would enable local government offices to provide prompt administrative services, the task of creating databases from conventional paper forms is expected to be time-consuming and costly.

According to a recent census, the current Japanese population is about 124 million, and there are more than 40 million family registers in Japan. Since there are about 3,200 local authorities handling family registers, a typical local authority would have to input about a hundred thousand family registers. The study group members conducted a joint survey of family registration documents in Toshima Ward, which showed that about 50% of existing documents are typed and the other 50% are handwritten, sometimes in cursive style. Thus, we have focused our efforts on developing a form image analysis system, which supports initial text entry of typed family registers. The results are registered in the database after several manual examination processes, along with other text data manually entered from handwritten forms.

This paper is organized as follows. In section 2, we describe an algorithm for analyzing the layouts of family register forms. Section 3 summarizes the flow of the process for entering family registration data, from image capturing to text recognition. Section 4 reports the performance of our system's layout analysis and character recognition, measured on actual images. Finally, the effectiveness of our system and future enhancements are discussed in section 5.

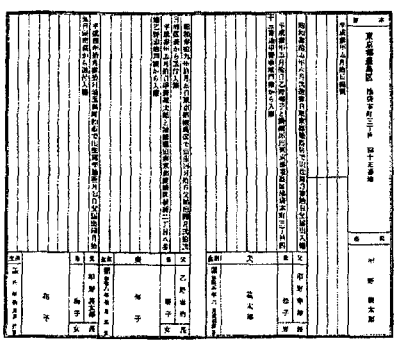


Fig. 1. An example of family register sheets.

2 Layout analysis of family register forms

A family register occupies several sheets of paper. Figure 1 shows an example of such sheets. Each sheet contains several rectangular fields segmented by horizontal and vertical ruled lines, and containing typed or handwritten text. We used a form layout analysis method to classify the layout and identify the fields to be recognized. A sheet image is segmented into two (left and right halves) images, which are analyzed separately. In this paper, we assume that the left and right images are pages. Observing family register forms page by page, we selected three major layouts to be distinguished. Although each layout is based on a fixed format, the overall sizes of the pages and the positions of fields on them vary according to the date of creation and the conditions of image capturing.

The method used for analyzing the layout of a family register consists of (1) detecting primitive blobs that represent character strings, vertical line segments, and horizontal line segments, and (2) matching predefined page models with the detected blobs.

1. Detection of primitive blobs[1]

The primitive blobs are detected by using information on the top and bottom boundaries of black components. Figure 2 shows the process flow of blob detection.

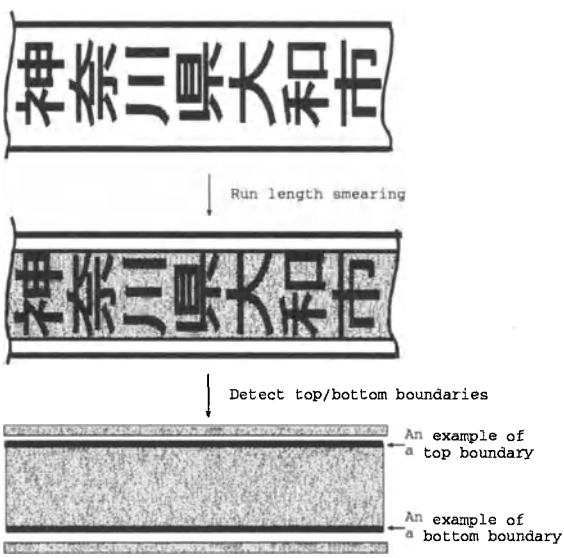


Fig. 2. A process flow of blob detection.

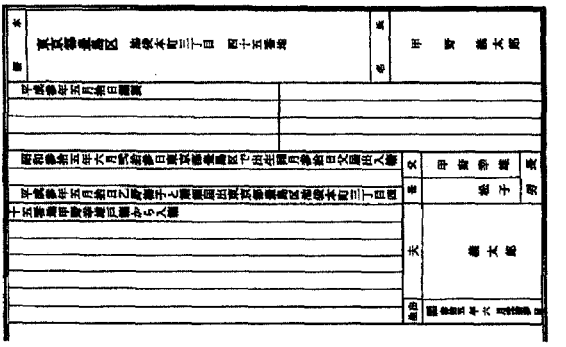


Fig. 3. An example of page images to be processed.

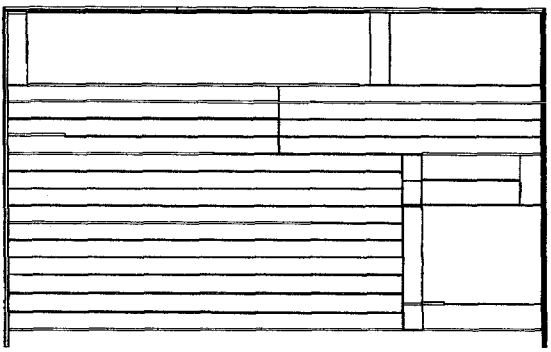
A page image rotated at an angle of 90 degrees is raster-scanned to generate smeared run-length data (starting positions and lengths) by replacing short horizontal white runs with black runs. Since characters are arranged horizontally in the rotated page image, separated black components that are constituents of a text line can be integrated by this smearing process.

Primitive blobs are detected as rectangles sandwiched between the top and bottom boundaries of the integrated components. These rectangles can be easily classified into character strings, vertical lines, and horizontal lines, because they maintain original height information. The top and bottom boundaries are detected by comparing the smeared run-length data from vertically successive lines. A run located under a white run is considered to be a part of a top boundary. Bottom boundaries can be detected in a corresponding manner. Figure 3 shows an original image (the right page of a sheet in Figure 1), and the results for this image are shown in Figure 4. Adjacent rectangles representing characters are then integrated.

The smearing process mentioned above sometimes connects characters to vertical ruled lines. If the connected regions are regarded as one component by a conventional component labeling technique, it is very difficult to divide this component into proper parts. Use of top and bottom boundary information can prevent such incorrect integrations.

2. Matching predefined page models with the detected blobs

Page models predefined for each form layout are matched with detected blobs (especially line segments) to determine the layout and identify the fields. A page model consists of three elements: (1) descriptions of ruled lines that distinguish the model from others, (2) descriptions of ruled lines that segment a page into specific fields, and (3) descriptions of field areas to be recognized. A description of a ruled line contains the approximate position and length of the line, and a flag indicating its angle (vertical or horizontal). It is not necessary to describe all the ruled lines in a page image. A description of a field contains identifiers of ruled lines enclosing the field.



(a) Detected lines

(b) Dtected character components

Fig. 4. Results of blob detection.

Figure 5 shows the flow of the matching process. A preprocess that connects lines located on the same straight line (if the gaps are less than a threshold value) is first applied, so that a matching program only has to check one-to-one correspondences between the ruled lines described in a page model and the detected lines.

Next, the matching program determines the correspondence of a model with detected lines by means of a depth-first search algorithm. This procedure starts by searching for a line that matches the first ruled line described in the model. A line is regarded as matching the corresponding ruled line when the following conditions are satisfied:

- The angle of the line is the same as that of the corresponding ruled line.
- The length of the line is the same as that of the corresponding ruled line within the range of a threshold value.
- The position of the line is almost the same as that of the corresponding ruled line.
- The topological relation (see Figure 6) between the line and a previously matched line is the same as that between two corresponding ruled lines.

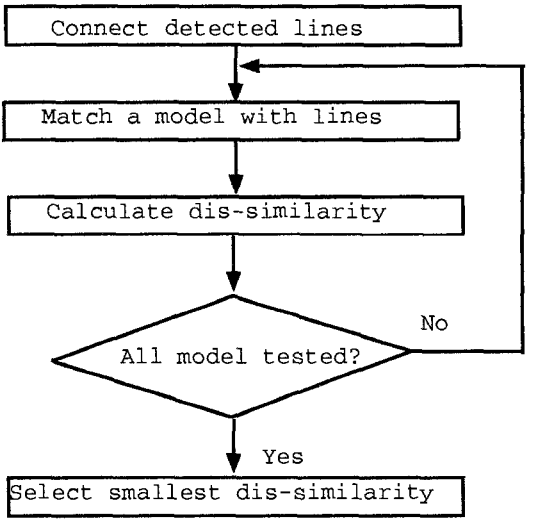


Fig. 5. A flowchart of matching process.

If a line is matched, it is recorded together with the corresponding ruled line, and a search for the next ruled line in the model is then performed. If no line is matched, the system backtracks to re-examine the search result for the most recently matched ruled line. The procedure terminates successfully when it finds matched lines for all ruled lines in the page model. Rectangular areas representing fields can be identified on the basis of the matched lines. This procedure is performed for all registered page models. If it fails to find a possible correspondence, the model is rejected. If multiple models match the detected lines, the model that gives the least dissimilarity is selected. The dissimilarity between the detected lines and a model is the sum of the dissimilarities between all the lines and the corresponding ruled lines in the model, which are calculated according to their differences in position and length.

This form layout analysis method does not depend on any specific knowledge or assumptions about family registration forms. It can be applied to other forms by modifying several parameters (e.g. the smearing threshold) and page models.

3 Data flow of family registration data entry

Figure 7 shows the basic hardware configuration of the family registration data entry system. The system consists of a microfilm scanner, a personal computer (PC) server for form layout analysis and character recognition, a laser printer, client PCs for manual data entry, and client PCs for verification and correction. An OCR adapter card which performs character recognition is attached in the PC server[4]. The task flow focused on in this paper is (1) scanning of microfilms, (2) form layout analysis, and (3) character recognition. Both the recognized data

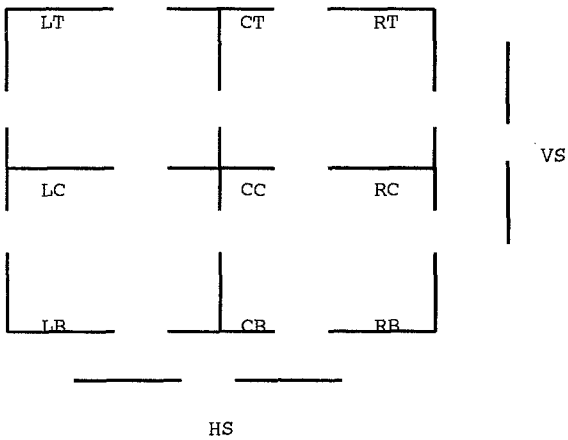


Fig. 6. Topological relations between two lines.

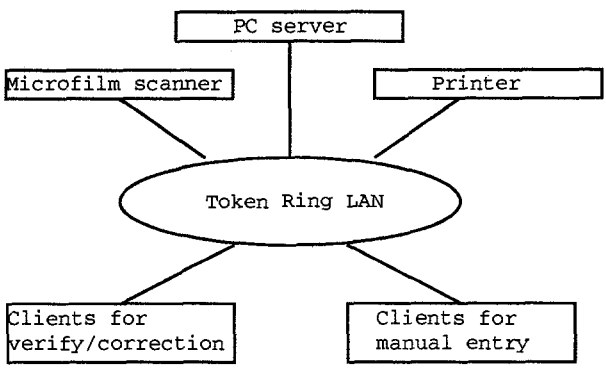


Fig. 7. Hardware configuration of family registration data entry system.

and manually entered data were examined and corrected by operators, and then converted into database format.

1. Scanning of microfilms

Since the original family registration forms must remain in the keeping of local authorities, all the forms are microfilmed on site, and later passed to the system. The microfilms are required to be preserved as backup. Binary images are captured by a microfilm scanner. The scanning resolution corresponds to 400 dots per inch for the original form. The bit images are printed and used for manual entry and later examination. Then, an operator classifies the images into those that can be automatically processed and those that must be manually entered.

2. Form layout analysis

A captured image is divided into two pages by using a projection histogram, and salt-and-pepper noises are then deleted. The form layout of each page is analyzed by the above method.

3. Character recognition

Characters in the identified field are segmented and recognized by a dedicated OCR adapter card. Since ruled lines which are close to characters sometimes cause segmentation/recognition errors, detected horizontal lines are erased on the image before segmentation process. Local contour direction features[2][3] are used to discriminate segmented character images. The recognition dictionary is generated from three typewriter fonts that are widely used for typing family registers. The number of categories to be recognized is about 3,000. The recognition result for each segmented image is a set of candidate codes and degrees of confidence. A sequence of character recognition results is then matched with a Japanese word dictionary, and the most likely path is selected[4][5]. The dictionary contains about 100,000 words (morphemes) and constraints on the transitions of words. We enhanced the dictionary with respect to place names and era names, because the family register description contains many birthplaces and dates of births.

4 Experiments

The accuracy of layout analysis and character recognition was estimated by using page images sampled from actual family register forms.

Six page models (including three backup models) were defined and matched with input images. A backup model is a simplified model consisting of ruled lines, which are indispensable for distinguishing three major layouts. Although the backup model has less ability to describe a layout (e.g. several distinct fields are identified as a single field), it can be matched even with an image in which some lines are patchy. When form layout analysis fails owing to lack of line

information, characters in fields are not recognized at all. Use of backup models reduces the workload of re-entering an entire text, because the characters can still be recognized according to the backup model.

The results of our first experiment are shown in Table 1. In this experiment, 96.2% of the layouts were correctly analyzed. In 11 cases out of 12, images for which the layout analysis failed were matched with one of the backup models. The failures resulted from lines not being detected because of image skewing.

Since the quality of a captured image varies according to its date of creation and other conditions, the recognition accuracy is not stable for all images. According to the measurements we obtained in another experiment, the accuracy for images of normal quality was 92.3%(see Table 2). Recognition errors were mainly caused by overlaps between seal impressions and characters, and by handwritten annotations.

Table 1. Accuracy of form layout analysis.

Total No. of Pages	317	
Success	305	96.2%
Failure	12	3.8%

Table 2. Accuracy of character recognition.

Total No. of Pages	27	
Total No. of Characters	5,527	
Correct	5,100	92.3%
Error(sub. and rejections)	427	7.7%

The average time needed for layout analysis was 10.4 seconds on a PC with a 80486 DX2 (66MHz) processor. The average time of character recognition including word matching was 13.4 seconds per page.

5 Concluding remarks

The family registration data entry system was actually used for initial entry of typed forms. It took about four months to process 137,000 family registrations, including both of typed and handwritten forms. On average, a family registration consists of 3.4 pages, which are the units processed by our system. According to an interview with the person in charge of the operation, about 75% of the typed forms were processed by using the system we developed, whereas the others were processed manually because their image quality was low or because they contained many handwritten corrections. Since requests for updating and addition

of family registration data were made during the period in which the data were being entered, a reduction of the time spent on data entry is a potential advantage of using the system. We think the system made a significant contribution to the creation of a family registration database.

We aim to improve the robustness of the system to deal with skewed images and handwritten annotations. Automatic classification of typed and handwritten forms is another requirement from an operational viewpoint. We think that these enhancements make it easier to use the system to create huge databases in a wide range of fields.

6 Acknowledgment

We would like thank Messrs. Shigeru Haijima and Isao Yabe in Toshima Ward for their guidance in the area of family registration, and our colleagues: Messrs. Daisuke Senba, Yuka Tateishi, Nobuyasu Itoh, Shin Katoh, Akio Yamashita, Hiroshi Maruyama, and Yuki Hirayama for their cooperation in implementing the system and for fruitful discussions.

References

1. Amano T. et al. : "A Character-String Detection Algorithm Using Horizontal Boundaries and Its Application to a Part-Number Entry System", Proc. SPIE, **Vol. 1,452**, pp.330-339 (1991).
2. Itoh N. and Takahashi H.: "A Hand-written Numeral Verification Method Using Distribution Maps of Structural Features",Proc. SPIE, **Vol. 1,258**, pp.150-159 (1990).
3. Takahashi H.: "A simple recognition method for handwritten Kanji characters by using primitive connective directions of thinning", Trans. IEICE Japan, **Vol. PRL82-8**, pp.57-62 (1982), in Japanese.
4. Amano T. et al. : "DRS: A Workstation-Based Document Recognition System for Text Entry", IEEE Computer, **25**, 7, pp.67-71 (1992).
5. Itoh N. and Maruyama H. : "A Method of Detecting Errors in the results of Japanese OCR", Trans. IPSJ , **33**, 5, pp.664-670 (1992), in Japanese.

Applying Handwriting Recognition to US Census Forms

Thomas M. Breuel

San Jose, CA, USA
tmb@best.com

1 Introduction

The US Bureau of the Census is considering automating part of the year 2000 Census effort using new computer technology. Traditionally, census data has been captured from census forms by keying it directly from the paper forms ("key from paper", KFP) and using special purpose optical mark readers for capturing information from checkboxes. The new technologies considered by the Census Bureau include:

- "key from image" (KFI) technology, i.e., presenting census workers with on-screen images of fields on census forms, rather than giving them paper forms
- optical mark recognition (OMR) from images, rather than using special purpose hardware on microfilm images of forms
- handwriting recognition (OCR) for capturing handwritten responses automatically

In order to gain experience with the application of these technologies, a smaller test census (the 1995 Test Census) is being conducted. This test involves real households, but it is limited to three cities. The 1995 Test Census incorporates KFP, KFI, OMR, and OCR technologies. KFP technologies are included to obtain a baseline for evaluating the performance of the other technologies.

Because its goals are different, the software configuration and workflow used for the 1995 Test Census differ significantly from that of an actual census. In the 1995 Test Census, KFP and KFI preceded OCR, and all handwritten fields were processed by all three methods[1]. Furthermore, quality control measures needed to identify potential problems during an ongoing actual census were limited.

The overall system was integrated and tested in collaboration between the US Bureau of the Census, the author (who provided the handwriting recognition part of the overall system), and several other participants. In this paper, we will examine a number of practical issues that have affected the handwriting recognition part of this effort.

[1] These differences in workflow may also have some impact on the measured performance of the different recognition methods. In particular, during key from paper operations, keyers sometimes mark up handwritten fields to disambiguate them. This artificially lowers the measured KFP and KFI error rates and higher OCR error rates.

The forms used in the 1995 Test Census correspond to the short forms used in actual US Census tasks. Responses on census forms are both given in the form of checkboxes and handwritten inputs. The set of handwritten fields present on 1995 Test Census forms is summarized in Table 1. Altogether, there are approximately 120 fields on each census form that can potentially contain handwritten input.

The evaluation of the handwriting recognition aspect of the 1995 Test Census is still an ongoing process. This paper summarizes some of the experiences and results of that effort[2].

2 Handwriting Recognition

The statistical and algorithmic basis of the handwriting recognition system used in the 1995 Test Census has been described elsewhere [1]. Basically, it is a system that takes binary images of complete handwritten responses as input and generates ASCII transcriptions. The system does not require forms based character segmentation information (i.e., no preprinted per-character boxes are needed). In past comparisons, recognition performance of this system has been found to be very competitive with other handwriting recognition systems for unsegmented handwritten fields [2]. For most fields, processing takes a few seconds per field on modern workstation hardware.

For the 1995 Test Census, a number of adaptations were implemented compared to the system used with the Second NIST/Census Handwriting Recognition Conference [2]:

- Originally, the system had only been designed for alphabetic data. The system was adapted to handle numerical data as well. This required basic changes to the character set code in the recognizer. It also required the implementation of a new segmentation algorithm better suited to the recognition of handwritten digits.

- Recognition times were reduced to a fraction of the original times through profiling, code optimization, and algorithmic improvements. In the original version of the system, recognition time was dominated by the running time of the Viterbi algorithm. In the current version of the system, most of the time is spent in the image processing stages.

In addition, a number of other algorithmic improvements were prototyped, including the use of classifier combination at the per-character recognition level, improved training methods, and improved segmentation schemes. On test cases, these changes resulted in significant decreases in error rates. Logistic constraints prevented the inclusion of these improvements in the production version of the handwriting recognizer.

[2]This paper represents the views of the author and not necessarily those of the U.S. Bureau of the Census.

Field	Type	Comment
first name	alpha	
last name	alpha	
middle initial	alpha	single letter
day of birth	num	2 digits
month of birth	num	2 digits
year of birth	num	2 or 4 digits
age	num	1 to 3 digits
race write-ins	alpha	present only on small fraction of forms
relationship write-in	alpha	present only on small fraction of forms

Table 1: Summary of field types present on 1995 Test Census forms.

3 Forms Design

In large volume applications, the design and layout of forms is constrained by a number of factors:

- "Respondent-friendly design" is an important concern in any large volume forms processing task, since non-responses or incomplete responses are often very costly. Determining reliably ahead of time whether a particular change to forms design will result in a significant increase in the number of non-responses in an actual application can be quite difficult.
- Available space is limited; changes to forms design that would necessitate the use of extra pages in the form are impossible.
- Changes to forms layout and design need to reviewed by several authorities.
- Changes to the content of forms may be imposed on relatively short notice.
- In large volume applications, multiple printing businesses may be used to print forms. This can result in some variation in the layout and ink used for individual forms.

Based on experience with handwriting recognition on existing US Census forms, a number of improvements to the forms design were suggested by the author, subject to the above constraints:

- The old forms design contained boxes printed in non-drop out ink surrounding each input field. In some fields, printed arrows protruded into the fields. Such printed forms background is impossible to remove reliably and interferes with handwriting recognition. The new forms design uses drop-out ink for delineating input fields.

- Some census forms designs use variable field heights. Since field height affects writing style, a handwriting recognizer trained on images of fields of a certain height will give poorer performance on images of fields of a different height. In the new forms design, field heights have been made consistent.

- Without changing overall space requirements, preprinted text and fields were modified slightly from the original forms design to achieve a consistent, minimal distance between handwritten input fields and surrounding preprinted text.

The use of preprinted boxes for individual letters and digits on numerical fields and name fields was also suggested. For numerical fields, such a change would probably not affect respondent friendliness negatively. Because of the many spelling variants that exist for names, the use of preprinted boxes is probably also very useful for achieving letter accurate transcriptions of names. The effect on respondent friendliness of using boxes for individual characters in name fields is less clear, however. Individuals with long names might not be able to fit their names into the moderate number of boxes for which there is room on the form.

No data is available to quantify the effects of these changes in forms design on the performance of the handwriting recognition system (this would have required sending out multiple designs to respondents). Based on a visual inspection of fields, it is clear, however, that these changes have had a significant positive effect on the quality of the input to the handwriting recognition system.

4 Integration of Recognizer

The 1995 Test Census effort is a large project, involving the integration of a complex paper based workflow, as well as several independently developed large software systems. System integration has been a collaborative and distributed effort, and installation and maintenance has involved multiple sites and locations. The complete system will be described by the Census Bureau elsewhere; here, we will focus on those aspects of system integration that directly affected the operation of the handwriting recognizer.

Because of the way the project was structured, opportunities for testing and on-site integration were very limited. This made the use of robust, easy to use interfaces between the handwriting recognizer and the rest of the system crucially important:

- The system is implemented under UNIX; UNIX memory protection, heavy-duty file system, unrestrictive file system namespace, and support for distributed computing proved to be very valuable for integration.

- All forms are identified using a unique serial number. All image files derived from a form have file names that are prefixed by this serial number, followed by a string that identifies the content of the image.

- A forms segmentation module (separate from the handwriting recognizer) preprocesses forms and generates binary images of isolated fields.
  These field images stored in Group-4 compressed TIFF files, with names of the form `<form_serial#>-<field_id>.tif`.
  The `<field_id>` part of the file name identifies which field on the form is represented by the TIFF file; for example, a `<field_id>` of `p03_rac_oth` identifies the TIFF file as the "race/other" write-in field for "Person 3" on the form.

- All the field images from a set ("batch") of 20 forms are collected and stored in a single compressed "tar" (UNIX "tape archive") file, the *batch file*. Also included in the batch file is the result of optical mark recognition (OMR), information that is used, for example, to determine whether a dictionary of male or female first names is to be used for recognizing the first name of a respondent. These batch files are entered into a queue.

- A queuing system removes batch files from the queue and distributes them to idle machines. On those machines, a server process creates a working directory, retrieves the individual field images from the batch file, and stores them in the working directory. It then starts up the handwriting recognizer in the working directory.

- The handwriting recognizer examines the directory it has been started up in. It looks for files with names that represent valid field names and proceeds to recognize them. The handwriting recognizer stores the results of the recognition process in a file called `batch_output` in the working directory. The `batch_output` file is a simple key/value pair file, containing for each field image a transcription, a confidence value, and a string identifying which language model was used by the handwriting recognizer to recognize the string (confidence values are not comparable among different models, so this information is needed later for choosing rejection thresholds for each model). It also creates a number of log files ending in the extension `.out` that can be examined to track problems.

- When the handwriting recognition process finishes, the queueing system will store the contents of `batch_output` in a commercial database holding all the results. The working directory and all its contents are removed by the queueing system.

This approach to integrating the handwriting recognizer has worked robustly and without problems. The use of batching has been crucial, because there is a significant startup cost involved in invoking the handwriting recognizer for each field type. The overhead involved in manipulating field images as TIFF files was determined to be small compared to the cost of handwriting recognition itself.

Initially, it had been planned to carry out one run of the complete system as a test run, in order to identify potential problems, and then do a second production run. However, the planned production run had to be canceled because

of financial constraints. Because a number of problems with training data and model selection were found during the test run (see below), error rates for the handwriting recognizer determined during that run may be significantly larger than actual rates of the system.

Interfacing with the database has required considerable programming effort, and the data stored inside the database is difficult to access for evaluations. Furthermore, the log files produced by the handwriting recognition system and the field images themselves are not stored in the database, limiting the amount of information accessible for performance evaluation. On the other hand, the primary functionalities provided by database systems, efficient selection, transaction processing, access control, and recovery, were not required in the 1995 Test Census task (or were already provided by other system components). System integration for similar efforts should carefully weigh the benefits and costs associated with using database technology. Even if the eventual users of the census data would like to use database technology for accessing and manipulating the results of handwriting recognition, it might be prudent not to integrate databases directly into the workflow for the recognition system.

5 Training

Experience with handwriting recognition systems shows that systems trained on a statistically representative sample of handwriting for a particular task can perform significantly better on that task than systems trained on some other data set. There are many possible explanations for this phenomenon, including the effects of changes in forms layout, response types, different writer populations, different preprocessing steps, and others. Even if a system has proven to work reasonably well across different tasks, for a large effort, it is still advisable to at least measure and confirm its performance on the new task.

In the context of the 1995 Test Census (and the year 2000 Census), this presents some logistic problems. Unlike tasks like the transcription of bank or medical forms, samples of census forms from actual respondents arrive over a very short period of time. In order to be usable as training data, some of those forms have to be transcribed. In addition, census applications usually impose strict constraints on when the output of the handwriting and keying processes need to be available for further processing (e.g., for congressional redistricting decisions).

In the 1995 Test Census, a period of about two weeks was allocated for training the handwriting recognition system. Since on the average only two workstations were available for training (a much smaller number than was available for handwriting recognition after training), this provided only very limited opportunity for training, model selection, and model validation. Training data was based on those census forms that were returned early. Transcription data was supplied in the form of output from the keying workstations. The handwriting recognition system was trained using this information[3].

[3] The training step had been tested during system integration using key from paper data.

As mentioned above, the only forms that were available for training were early returns. It is likely that the writing style and quality changes for census forms returned at different times, both due to geographic variation, and due to different respondent attitudes[4]. Another factor that can affect handwriting recognition over time is changes in scanner parameters. After about two thirds of the census forms had been scanned (only a single high-volume scanner has been used for the 1995 Test Census), the scanner appears to have developed a defect, and scanner settings may have been changed in an attempt to compensate. The effects of this on handwriting recognition still need to be determined.

Of course, we had been aware of the need for training and performance monitoring during an actual census from the beginning; the 1995 Test Census was not designed to be a complete rehearsal of an actual census. But based on the practical experience, the following points should be noted:

- Testing, and possibly training, the system on actual data is an important step and needs to be considered when planning future census data capture efforts. Sufficient computational resources need to be available to allow training to be done quickly.
- On-line monitoring of performance is important to be able to detect and possible react to changes in writing styles and scanner problems.

6 OMR Information

A specific adaptation to the census task, implemented in the handwriting recognizer, is the use of optical mark recognition (OMR) information. On each census form, respondents usually indicated their sex and race by checking one of a number of checkboxes. These checkboxes can be recognized by computer with high reliability and can be used to advantage for improving the performance of handwriting recognition.

In the current system, the use of OMR information was investigated for two purposes: (1) to select different language models for female and male first names, and (2) to select different language models for first and last names depending on the race of the respondent.

The use of OMR information is limited by the data that is available for building language models. For example, in order to take advantage of the information

When the actual training for the 1995 Test Census was about to begin, it became clear that the format of the transcription data made available by the manual keying operations had changed, and software had to be built quickly to use the new format for training. As it turned out later, that software had a minor bug, resulting in a few percent of the numerical training data being mislabeled. Because of the constraints on time and computational resources, there was no opportunity to restart training when this was discovered. Because the handwriting recognizer is designed to reject some kinds of incorrect training data, this has probably had only a limited effect on error rates for numerical fields, but the actual impact still needs to be determined.

[4]This also means that the data available for training on the 1995 Test Census task may not have been a statistically representative sample of the images actually encountered by the handwriting recognition system during of processing.

that a particular individual is an "Asian-Pacific Islander" (API), a statistically representative sample of last names for Asian-Pacific Islanders resident in the US has to be available. Without such a sample, the benefits of reductions in the perplexity of language model through the use of OMR information may be outweighed by the problems associated with lower coverage of the language model.

Making the optimal tradeoffs between coverage and perplexity for language models requires an evaluation and comparison of the different models on the actual task (model selection). Given the limited amount of time available, only models taking advantage of OMR information about a respondent's sex could be validated for use with the 1995 Test Census, and reductions in error rate were significant. Preliminary data suggests that taking advantage of race information for selecting separate Hispanic/non-Hispanic language models for first and last names may also be beneficial.

7 Inter-Field Constraints

Forms sometimes contain redundant information. The forms used for the 1995 Test Census contained two kinds of redundant information:

- respondents wrote their names twice, once on a "roster page", and a second time in the area where they provided additional personal information
- each respondent provided both their age and their date of birth

Commonly, such information is taken advantage of for the purposes of handwriting recognition by performing consistency checks after individual fields have been recognized.

The handwriting recognition system used in this task took a different approach: the optimal interpretation of the input was found by recognizing related input fields simultaneously and imposing the consistency constraints as part of the language model. For the repeated name fields, the new language model consists simply of the concatenation of the language model for a single name field with itself. Imposing the constraint on the relationship between date of birth and age in general would require a complex language model. However, the set of possible age responses is small, so that a finite approximation is sufficient.

The case of imposing the date-of-birth/age constraints was validated during the brief period where real task data was available and resulted in significant improvements in performance. Time did not permit validation of the language model that took advantage of the presence of repetitions of names on the forms.

8 Discussion

The goal of the 1995 Test Census has been to gather more experience with the application of KFI, OMR, and OCR technologies to census tasks. This will allow the Census Bureau to plan ahead better for the year 2000 Census.

From the perspective of the handwriting recognition (OCR) part of the 1995 Test Census, the following issues should receive particular consideration in future census efforts incorporating handwriting recognition:

- On-going quality control of handwriting recognizers needs to be integrated into the workflow of an actual census task, just like on-going quality control of manual entry is already in place. Non-stationary distributions of writing styles as well as problems with printing and scanning could significantly affect OCR performance. To compensate, handwriting recognizers may need to be retrained on-line. Human and computational resources need to be budgeted for this purpose.

- To support quality control and troubleshooting for the OCR subsystem (and probably for KFI as well), image and log file information needs to be readily available.

- Forms design is crucially important. In particular, the use of individual boxes for characters in numerical fields and possibly name fields should be investigated. (A quantification of this effect will have to await a more detailed classification of the OCR errors.)

- OMR information and redundant fields have the potential for significantly improving OCR performance. Any decisions on workflow should take this into account.

More specific recommendations will have to wait until more performance data from the 1995 Test Census effort is available. Only preliminary measurements of the performance and throughput of the handwriting recognition system are available so far. Errors still need to be reconciled between KFP, KFI, and OCR results, and error sources still need to be identified.

References

[1] Thomas M. Breuel. Design and Implementation of a System for the Recognition of Handwritten Responses on US Census Forms, 1994. Document Analysis Systems Workshop 1994, Kaiserslautern, Germany.

[2] *The Second Census Optical Character Recognition System Conference*, February 1994.

Author Index

Aggarwal, J.K. 43, 113
Aggarwal, S.J. 43
Ahuja, N. 139
Aloimonos, Y. 33
Amano, T. 583
Ando, H. 467
Aubert, G. 71

Baker, K. 519
Bober, M. 91
Bobick, A. 23
Bolle, R. 33
Bouthemy, P. 101
Breuel, T.M. 447, 593
Bunke, H. 373

Charan, R. 139
Cheng, T.-K. 219
Chien, C.-H. 59
Christensen, H.I. 201
Curwen, R. 425

Davies, A.C. 489
Deriche, R. 71
Dickmanns, E.D. 499

Ebihara, K. 159
Eklundh, J.-O. 181

Fermüler, C. 33
Fiala, J.C. 171
Forsyth, D. 425
Frankel, C.H. 551

Gehrke, A. 81

Haag, M. 81
Hancock, E.R. 353
He, Z. 129
Hild, M. 479
Horstmann, J. 201
Hu, Z. 231
Huang, T.S. 13
Hudson, M. 289

Jain, R. 571

Kishino, F. 159
Kita, N. 191
Kitchen, L. 219
Kittler, J. 91, 383
Klauck, C. 561
Kluge, K. 329
Kollnig, H. 509
Kornprobst, P. 71
Kuniyoshi, Y. 191

Lakshmanan, S. 329
Lee, C.-H. 149
Lee, J.C.M. 541
Li, S.Z. 363
Liao, W.-H. 43
Lindenbaum, M. 457
Liou, M.L. 129
Liu, J. 415, 425

Liu, L.Y. 255
Liu, Z.-Q. 219
Lopez, R. 13
Lu, M. 551

Ma, R.H. 541
Ma, S.D. 231
Maki, A. 181
Mano, T. 583
Mardia, K.V. 297
Messmer, B.T. 373
Moons, T. 405
Mundy, J. 415, 425

Nagel, H.-H. 81, 509
Nordlund, P. 181

Odobez, J.-M. 101
Ohya, J. 159
Otte, M. 81

Papachristou, P. 383
Pelillo, M. 343
Petrou, M. 383
Picard, R. 3
Poggio, T. 437
Pollefeys, M. 405

Rasmussen, T. 201
Rothwell, C. 425
Rougeaux, S. 191

Santini, S. 571
Sato, Y. 211
Schneiderman, H. 171
Shah, S. 113
Shashua, A. 395
Shen, D.G. 541
Shirai, Y. 479
Suehiro, T. 191
Sullivan, G.D. 519
Sung, K.-K. 437
Swain, M.J. 551

Tan, T.N. 519
Thomanek, F. 499
Toriyama, S. 583
Toyokawa, K. 583

Uhlin, T. 181

Van Gool, L. 405
Velastin, S.A. 489

Walker, E. 415
Wang, C. 231
Wang, J.-S. 307
Wavering, A. 171
Wilson, R.C. 353
Won, C.S. 319

Xie, M. 255
Xiong, W. 541
Xu, G. 265

Yin, J.H. 489

Zhang, H.-J. 531
Zhang, Z. 241
Zheng, J.-Y. 275
Zisserman, A. 425

Lecture Notes in Computer Science

For information about Vols. 1–957

please contact your bookseller or Springer-Verlag

Vol. 958: J. Calmet, J.A. Campbell (Eds.), Integrating Symbolic Mathematical Computation and Artificial Intelligence. Proceedings, 1994. X, 275 pages. 1995.

Vol. 959: D.-Z. Du, M. Li (Eds.), Computing and Combinatorics. Proceedings, 1995. XIII, 654 pages. 1995.

Vol. 960: D. Leivant (Ed.), Logic and Computational Complexity. Proceedings, 1994. VIII, 514 pages. 1995.

Vol. 961: K.P. Jantke, S. Lange (Eds.), Algorithmic Learning for Knowledge-Based Systems. X, 511 pages. 1995. (Subseries LNAI).

Vol. 962: I. Lee, S.A. Smolka (Eds.), CONCUR '95: Concurrency Theory. Proceedings, 1995. X, 547 pages. 1995.

Vol. 963: D. Coppersmith (Ed.), Advances in Cryptology - CRYPTO '95. Proceedings, 1995. XII, 467 pages. 1995.

Vol. 964: V. Malyshkin (Ed.), Parallel Computing Technologies. Proceedings, 1995. XII, 497 pages. 1995.

Vol. 965: H. Reichel (Ed.), Fundamentals of Computation Theory. Proceedings, 1995. IX, 433 pages. 1995.

Vol. 966: S. Haridi, K. Ali, P. Magnusson (Eds.), EURO-PAR '95 Parallel Processing. Proceedings, 1995. XV, 734 pages. 1995.

Vol. 967: J.P. Bowen, M.G. Hinchey (Eds.), ZUM '95: The Z Formal Specification Notation. Proceedings, 1995. XI, 571 pages. 1995.

Vol. 968: N. Dershowitz, N. Lindenstrauss (Eds.), Conditional and Typed Rewriting Systems. Proceedings, 1994. VIII, 375 pages. 1995.

Vol. 969: J. Wiedermann, P. Hájek (Eds.), Mathematical Foundations of Computer Science 1995. Proceedings, 1995. XIII, 588 pages. 1995.

Vol. 970: V. Hlaváč, R. Šára (Eds.), Computer Analysis of Images and Patterns. Proceedings, 1995. XVIII, 960 pages. 1995.

Vol. 971: E.T. Schubert, P.J. Windley, J. Alves-Foss (Eds.), Higher Order Logic Theorem Proving and Its Applications. Proceedings, 1995. VIII, 400 pages. 1995.

Vol. 972: J.-M. Hélary, M. Raynal (Eds.), Distributed Algorithms. Proceedings, 1995. XI, 333 pages. 1995.

Vol. 973: H.H. Adelsberger, J. Lažanský, V. Mařík (Eds.), Information Management in Computer Integrated Manufacturing. IX, 665 pages. 1995.

Vol. 974: C. Braccini, L. DeFloriani, G. Vernazza (Eds.), Image Analysis and Processing. Proceedings, 1995. XIX, 757 pages. 1995.

Vol. 975: W. Moore, W. Luk (Eds.), Field-Programmable Logic and Applications. Proceedings, 1995. XI, 448 pages. 1995.

Vol. 976: U. Montanari, F. Rossi (Eds.), Principles and Practice of Constraint Programming — CP '95. Proceedings, 1995. XIII, 651 pages. 1995.

Vol. 977: H. Beilner, F. Bause (Eds.), Quantitative Evaluation of Computing and Communication Systems. Proceedings, 1995. X, 415 pages. 1995.

Vol. 978: N. Revell, A M. Tjoa (Eds.), Database and Expert Systems Applications. Proceedings, 1995. XV, 654 pages. 1995.

Vol. 979: P. Spirakis (Ed.), Algorithms — ESA '95. Proceedings, 1995. XII, 598 pages. 1995.

Vol. 980: A. Ferreira, J. Rolim (Eds.), Parallel Algorithms for Irregularly Structured Problems. Proceedings, 1995. IX, 409 pages. 1995.

Vol. 981: I. Wachsmuth, C.-R. Rollinger, W. Brauer (Eds.), KI-95: Advances in Artificial Intelligence. Proceedings, 1995. XII, 269 pages. (Subseries LNAI).

Vol. 982: S. Doaitse Swierstra, M. Hermenegildo (Eds.), Programming Languages: Implementations, Logics and Programs. Proceedings, 1995. XI, 467 pages. 1995.

Vol. 983: A. Mycroft (Ed.), Static Analysis. Proceedings, 1995. VIII, 423 pages. 1995.

Vol. 984: J.-M. Haton, M. Keane, M. Manago (Eds.), Advances in Case-Based Reasoning. Proceedings, 1994. VIII, 307 pages. 1995.

Vol. 985: T. Sellis (Ed.), Rules in Database Systems. Proceedings, 1995. VIII, 373 pages. 1995.

Vol. 986: Henry G. Baker (Ed.), Memory Management. Proceedings, 1995. XII, 417 pages. 1995.

Vol. 987: P.E. Camurati, H. Eveking (Eds.), Correct Hardware Design and Verification Methods. Proceedings, 1995. VIII, 342 pages. 1995.

Vol. 988: A.U. Frank, W. Kuhn (Eds.), Spatial Information Theory. Proceedings, 1995. XIII, 571 pages. 1995.

Vol. 989: W. Schäfer, P. Botella (Eds.), Software Engineering — ESEC '95. Proceedings, 1995. XII, 519 pages. 1995.

Vol. 990: C. Pinto-Ferreira, N.J. Mamede (Eds.), Progress in Artificial Intelligence. Proceedings, 1995. XIV, 487 pages. 1995. (Subseries LNAI).

Vol. 991: J. Wainer, A. Carvalho (Eds.), Advances in Artificial Intelligence. Proceedings, 1995. XII, 342 pages. 1995. (Subseries LNAI).

Vol. 992: M. Gori, G. Soda (Eds.), Topics in Artificial Intelligence. Proceedings, 1995. XII, 451 pages. 1995. (Subseries LNAI).

Vol. 993: T.C. Fogarty (Ed.), Evolutionary Computing. Proceedings, 1995. VIII, 264 pages. 1995.

Vol. 994: M. Hebert, J. Ponce, T. Boult, A. Gross (Eds.), Object Representation in Computer Vision. Proceedings, 1994. VIII, 359 pages. 1995.

Vol. 995: S.M. Müller, W.J. Paul, The Complexity of Simple Computer Architectures. XII, 270 pages. 1995.

Vol. 996: P. Dybjer, B. Nordström, J. Smith (Eds.), Types for Proofs and Programs. Proceedings, 1994. X, 202 pages. 1995.

Vol. 997: K.P. Jantke, T. Shinohara, T. Zeugmann (Eds.), Algorithmic Learning Theory. Proceedings, 1995. XV, 319 pages. 1995.

Vol. 998: A. Clarke, M. Campolargo, N. Karatzas (Eds.), Bringing Telecommunication Services to the People – IS&N '95. Proceedings, 1995. XII, 510 pages. 1995.

Vol. 999: P. Antsaklis, W. Kohn, A. Nerode, S. Sastry (Eds.), Hybrid Systems II. VIII, 569 pages. 1995.

Vol. 1000: J. van Leeuwen (Ed.), Computer Science Today. XIV, 643 pages. 1995.

Vol. 1001: M. Sudan, Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems. XIV, 87 pages. 1995.

Vol. 1002: J.J. Kistler, Disconnected Operation in a Distributed File System. XIX, 249 pages. 1995.

VOL. 1003: P. Pandurang Nayak, Automated Modeling of Physical Systems. XXI, 232 pages. 1995. (Subseries LNAI).

Vol. 1004: J. Staples, P. Eades, N. Katoh, A. Moffat (Eds.), Algorithms and Computation. Proceedings, 1995. XV, 440 pages. 1995.

Vol. 1005: J. Estublier (Ed.), Software Configuration Management. Proceedings, 1995. IX, 311 pages. 1995.

Vol. 1006: S. Bhalla (Ed.), Information Systems and Data Management. Proceedings, 1995. IX, 321 pages. 1995.

Vol. 1007: A. Bosselaers, B. Preneel (Eds.), Integrity Primitives for Secure Information Systems. VII, 239 pages. 1995.

Vol. 1008: B. Preneel (Ed.), Fast Software Encryption. Proceedings, 1994. VIII, 367 pages. 1995.

Vol. 1009: M. Broy, S. Jähnichen (Eds.), KORSO: Methods, Languages, and Tools for the Construction of Correct Software. X, 449 pages. 1995. Vol.

Vol. 1010: M. Veloso, A. Aamodt (Eds.), Case-Based Reasoning Research and Development. Proceedings, 1995. X, 576 pages. 1995. (Subseries LNAI).

Vol. 1011: T. Furuhashi (Ed.), Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms. Proceedings, 1994. (Subseries LNAI).

Vol. 1012: M. Bartošek, J. Staudek, J. Wiedermann (Eds.), SOFSEM '95: Theory and Practice of Informatics. Proceedings, 1995. XI, 499 pages. 1995.

Vol. 1013: T.W. Ling, A.O. Mendelzon, L. Vieille (Eds.), Deductive and Object-Oriented Databases. Proceedings, 1995. XIV, 557 pages. 1995.

Vol. 1014: A.P. del Pobil, M.A. Serna, Spatial Representation and Motion Planning. XII, 242 pages. 1995.

Vol. 1015: B. Blumenthal, J. Gornostaev, C. Unger (Eds.), Human-Computer Interaction. Proceedings, 1995. VIII, 203 pages. 1995.

VOL. 1016: R. Cipolla, Active Visual Inference of Surface Shape. XII, 194 pages. 1995.

Vol. 1017: M. Nagl (Ed.), Graph-Theoretic Concepts in Computer Science. Proceedings, 1995. XI, 406 pages. 1995.

Vol. 1018: T.D.C. Little, R. Gusella (Eds.), Network and Operating Systems Support for Digital Audio and Video. Proceedings, 1995. XI, 357 pages. 1995.

Vol. 1019: E. Brinksma, W.R. Cleaveland, K.G. Larsen, T. Margaria, B. Steffen (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. Selected Papers, 1995. VII, 291 pages. 1995.

Vol. 1020: I.D. Watson (Ed.), Progress in Case-Based Reasoning. Proceedings, 1995. VIII, 209 pages. 1995. (Subseries LNAI).

Vol. 1021: M.P. Papazoglou (Ed.), OOER '95: Object-Oriented and Entity-Relationship Modeling. Proceedings, 1995. XVII, 451 pages. 1995.

Vol. 1022: P.H. Hartel, R. Plasmeijer (Eds.), Functional Programming Languages in Education. Proceedings, 1995. X, 309 pages. 1995.

Vol. 1023: K. Kanchanasut, J.-J. Lévy (Eds.), Algorithms, Concurrency and Knowlwdge. Proceedings, 1995. X, 410 pages. 1995.

Vol. 1024: R.T. Chin, H.H.S. Ip, A.C. Naiman, T.-C. Pong (Eds.), Image Analysis Applications and Computer Graphics. Proceedings, 1995. XVI, 533 pages. 1995.

Vol. 1025: C. Boyd (Ed.), Cryptography and Coding. Proceedings, 1995. IX, 291 pages. 1995.

Vol. 1026: P.S. Thiagarajan (Ed.), Foundations of Software Technology and Theoretical Computer Science. Proceedings, 1995. XII, 515 pages. 1995.

Vol. 1027: F.J. Brandenburg (Ed.), Graph Drawing. Proceedings, 1995. XII, 526 pages. 1996.

Vol. 1028: N.R. Adam, Y. Yesha (Eds.), Electronic Commerce. X, 155 pages. 1996.

Vol. 1029: E. Dawson, J. Golić (Eds.), Cryptography: Policy and Algorithms. Proceedings, 1995. XI, 327 pages. 1996.

Vol. 1030: F. Pichler, R. Moreno-Díaz, R. Albrecht (Eds.), Computer Aided Systems Theory - EUROCAST '95. Proceedings, 1995. XII, 539 pages. 1996.

Vol.1031: M. Toussaint (Ed.), Ada in Europe. Proceedings, 1995. XI, 455 pages. 1996.

Vol. 1032: P. Godefroid, Partial-Order Methods for the Verification of Concurrent Systems. IV, 143 pages. 1996.

Vol. 1033: C.-H. Huang, P. Sadayappan, U. Banerjee, D. Gelernter, A. Nicolau, D. Padua (Eds.), Languages and Compilers for Parallel Computing. Proceedings, 1995. XIII, 597 pages. 1996.

Vol. 1034: G. Kuper, M. Wallace (Eds.), Constraint Databases and Applications. Proceedings, 1995. VII, 185 pages. 1996.

Vol. 1035: S.Z. Li, D.P. Mital, E.K. Teoh, H. Wang (Eds.), Recent Developments in Computer Vision. Proceedings, 1995. XI, 604 pages. 1996.