Digital Representations of the Real World

HOW TO CAPTURE, MODEL, AND RENDER VISUAL REALITY

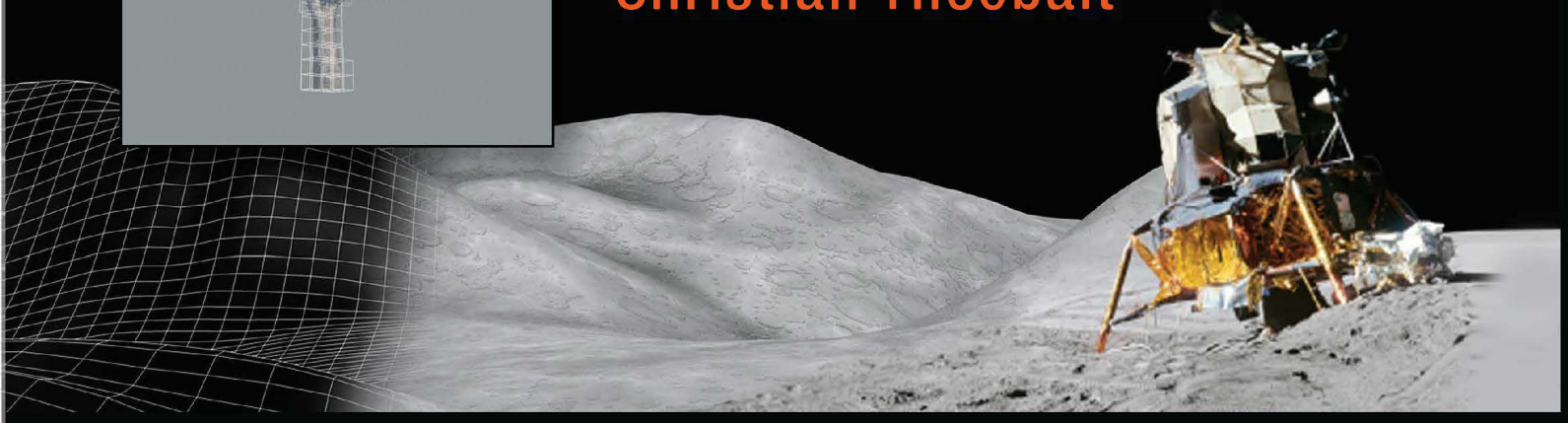
EDITED BY

Marcus A. Magnor

Oliver Grau

Olga Sorkine-Hornung

Christian Theobalt



CRC Press

Taylor & Francis Group

AN A K PETERS BOOK

Digital Representations of the Real World

HOW TO CAPTURE, MODEL, AND RENDER VISUAL REALITY

EDITED BY

Marcus A. Magnor
Oliver Grau
Olga Sorkine-Hornung
Christian Theobalt

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20141204

International Standard Book Number-13: 978-1-4822-4381-9 (Hardback)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Foreword

Over the last two decades, the realism achievable using computer graphics has increased to the point where it is now impossible, in visual effects applications, to distinguish computer-generated imagery from reality. Over the same time period, our ability to capture and re-synthesize the real world inside the computer has kept pace. These techniques enable us to quickly and accurately capture 3D objects and scenes with a high degree of geometric and photometric fidelity.

Examples of such capture systems include active range scanning, most notably affordable real-time depth cameras such as Kinect™. They also include passive image-based modeling algorithms, which take as input collections of regular RGB images or videos and produce 3D shape and appearance models. Recent examples of such systems include the research Photo Tourism system, the consumer-level Photosynth Web service, as well as 3D image-based capture systems such as 123D® Catch.

This book, *Digital Representations of the Real World: How to Capture, Model, and Render Visual Reality*, contains a comprehensive compendium of the myriad techniques that enable us to capture, model, and render the world with a high degree of realism. It reviews the variety of sensors, such as regular cameras, wide-angle omnidirectional cameras, active range scanners, and plenoptic (multi-viewpoint) cameras, used to capture 3D scenes, as well as fundamental algorithms, such as 3D structure and motion recovery and stereo correspondence, used to process this sensed imagery.

The book also describes 3D modeling techniques, including both generic object models such as 3D meshes, and more domain-specific models such as human shape and motion models, needed to efficiently capture and manipulate 3D scenes. Finally, it describes how these shape and appearance models can be rendered in a way that meets both speed (e.g., real-time interactivity) and realism requirements, often using techniques such as image- and video-based rendering and incorporating modern models of visual perception and fidelity.

The scope and breadth of the techniques and systems used to capture, model, and render realistic simulacra of 3D scenes are quite daunting and

can be a challenge for newcomers. This book provides an excellent introduction to and survey of this diverse field, written by some of the foremost researchers and practitioners in the field. Whether you are a novice to this exciting and challenging area, or an experienced veteran working in this field, you are sure to discover a wealth of useful and inspiring information in these pages.

Please dive in and enjoy!

Richard Szeliski
Microsoft Research

Preface

Marcus Magnor, Oliver Grau, Olga Sorkine-Hornung, and Christian Theobalt

Reality: The final frontier. Since the early beginnings of computer graphics, creating authentic models of real-world objects and achieving visual realism have been major goals in graphics research. Over the years, ingenious ways have been devised to represent real objects digitally, to efficiently simulate and emulate the laws of optics and physics, and to re-create perceptually authentic appearance. Ever-increasing CPU and GPU performance paved the way, up to the point where the memory and computational power available today afford genuine visual realism.

With visual realism within reach of modern hard- and software, intriguing new computer graphics applications have become possible. By combining computer graphics methods with video acquisition technology and computer vision algorithms, real-world events can now be interactively explored and experienced from an arbitrary perspective, almost like a video game. At the same time, the pursuit of visual realism has created new challenges. Higher visual realism can be achieved only from more detailed and accurate scene models. Consequently, the modeling process has become the limiting factor in attaining visual realism. Following the traditional paradigm, the manual creation of digital models consisting of 3D object geometry and texture, surface reflectance characteristics and scene illumination, character motion and emotion is a very labor-intensive, tedious process. The cost of conventionally creating models of sufficient complexity to engage the full potential of modern graphics hard- and software increasingly threatens to stall further progress in computer graphics.

To overcome this bottleneck, an increasing number of researchers and engineers worldwide have started to investigate alternative approaches in how to create digital models directly and automatically from real-world objects and scenes, with encouraging results: By now, entire cities are being digitized using panorama video footage, 3D scanners, and GPS; from CAD data and measured surface reflectance characteristics, highly realistic

digital mock-ups of prototypes are being created, e.g., for the automotive industry; algorithms are being developed to create stereoscopic movies from standard, monocular footage; and live TV sports broadcasts are being augmented in real-time with computer graphics annotations. Other graphics application areas that work on merging the real with virtual worlds are special effects production for movies and computer games. In their goal to construct convincing virtual environments and digital actors, special effects production companies heavily rely on techniques to capture models from the real world. Still, a lot of time must be spent on manual post-processing and modeling. As an alternative approach, the computer graphics and vision communities are working on image- and video-based scene reconstruction approaches that can capture richer and more complex models of objects, humans, and entire complex scenes.

The trend toward model capture from real-world examples is also being pushed by new sensor technologies becoming available at mass-market prices. Microsoft's Kinect™ depth cameras, Lytro's light field cameras, Point Grey's Ladybug™ omni-directional cameras, and other companies' products offer unprecedented, novel ways to capture the appearance as well as other attributes of real-world objects and events. Finally, the pervasiveness of smartphones containing video chips, GPS, orientation sensors, and more gadgetry may in the near future lead to new real-world capture paradigms based on swarms of networked handheld devices.

Robust methods to unobtrusively capture comprehensive digital models of the real-world are one important part for attaining visual realism in computer graphics. Still, model reconstruction from real-world captured data remains, in general, an ill-posed problem that is prone to errors and failure cases. Insight into our human visual perception, however, allows for developing new model-adaptive, perception-aware rendering approaches that are able to perceptually mask and conceal modeling error–induced visual artifacts. Investigating how to best integrate new capture modalities, reconstruction approaches, and visual perception into the computer graphics pipeline, or how to alter the traditional graphics pipeline to make optimal use of the many new possibilities, has become a top priority in computer graphics.

The following 23 chapters present the state-of-the-art of how to create visual realism in computer graphics from the real world. A total of 48 authors from all over the world have joined up to compile a comprehensive overview, covering in 5 parts the entire pipeline from acquisition, reconstruction, and modeling to realistic rendering and applications. While editing the book, we tried to strike a balance between a general, comprehensive introduction to this exciting new research area and a practical guide that shows how to get started on re-implementing and using many of the most frequently encountered methods. We hope that it will be helpful to

graduate students as well as researchers in academia and industry who are working in computer graphics, computer vision, multimedia, or image communications and who want to start their own research experiments in the challenging new field of real-world visual computing.

For MATLAB® and Simulink® product information, please contact:
The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA, 01760-2098 USA
Tel: 508-647-7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com

Contributors

Johannes Behr Fraunhofer IGD

Philippe Bekaert Hasselt University

Tamy Boubekeur Telecom ParisTech–CNRS LTCI–Institut Mines-Telecom, Paris, France

Edmond Boyer INRIA Grenoble Rhône-Alpes

Dan Casas University of Surrey–Centre for Vision, Speech and Signal Processing

Darren Cosker Department of Computer Science, University of Bath

Carsten Dachsbacher Karlsruhe Institute of Technology

Qionghai Dai Tsinghua University

Robert Dawes BBC Research and Development

Edilson de Aguiar CEUNES/UFES in São Mateus

Enrique Dunn The University of North Carolina at Chapel Hill, USA

Elmar Eisemann Delft University of Technology

Martin Eisemann TU Braunschweig, University of Technology

Peter Eisert Humboldt Universität zu Berlin, Fraunhofer HHI

Dieter W. Fellner Fraunhofer IGD / TU Darmstadt

Jan-Michael Frahm University of North Carolina at Chapel Hill, USA

Martin Fuchs University of Stuttgart

Juergen Gall Bonn University

Leonid German Institut für Informationsverarbeitung, Leibniz Universität Hannover

Bastian Goldlücke University of Konstanz, Department of Computer and Information Science

Oliver Grau Intel Visual Computing Institute

Volker Helzle Institute of Animation, Visual Effects and Digital Postproduction at Filmakademie Baden-Wuerttemberg

Anna Hilsmann Humboldt-Universität zu Berlin, Fraunhofer HHI

Adrian Hilton University of Surrey – Centre for Vision, Speech and Signal Processing

Peng Huang University of Surrey–Centre for Vision, Speech and Signal Processing

Muhannad Ismaël Université de Reims Champagne Ardenne

Jan Kautz NVIDIA Corporation

Oliver Klehm MPI Informatik

Thomas Knop Stargate Germany

Andreas Kolb University Siegen

Hendrik P. A. Lensch University of Tübingen

Max Limper Fraunhofer IGD/TU Darmstadt

Christian Lipski Metaio GmbH

Yebin Liu Tsinghua University, China

Céline Loscos University of Reims Champagne-Ardenne

Laurent Lucas Université de Reims Champagne-Ardenne

Fabrizio Pece ETH Zurich, Department of Computer Science

Yannick Remion Université de Reims Champagne Ardenne, France

Lorenz Rogge TU Braunschweig, University of Technology

Bodo Rosenhahn Institut für Informationsverarbeitung, Leibniz Universität Hannover

Kai Ruhl TU Braunschweig, University of Technology

Holly Rushmeier Yale University, USA

Michael Stengel TU Braunschweig, University of Technology

Graham Thomas BBC Research and Development

Kiran Varanasi Technicolor Research

Sven Wanner Heidelberg Collaboratory for Image Processing

Ruigang Yang University of Kentucky, USA

Jiejie Zhu SRI International

Stefanie Wuhrer Cluster of Excellence on Multimodal Computing and Interaction, Saarland University

Image Credits

Figure 1.3 Images courtesy of Philippe Bekaert at Hasselt University, Belgium, EU "2020 3D Media" project

Figure 2.2 Images courtesy of BBC, UK

Figure 3.2 Images courtesy of Philippe Bekaert at Hasselt University, Belgium, iMinds "explorative television" project

Figure 3.3 Image courtesy of Philippe Bekaert at Hasselt University, Belgium

Figure 3.4 Image courtesy of Philippe Bekaert at Hasselt University and Eric Joris at CREW vzw, "ICoSOLE" and "DreamSpace" EU projects

Figure 4.2 Image credits [Raposo et al. 13]

Figure 4.3 Image credits [Butler et al. 12] - accompanying video

Figure 4.4 Image courtesy of [Kolb et al. 10], Eurographics Association, 2010

Figure 4.5 Image courtesy of Left pmdtechnologies GmbH

Figure 4.7 Image courtesy of [Kolb et al. 10], Eurographics Association, 2010

Figure 4.8 Image courtesy of [Lefloch et al. 13], SPIE, 2013

Figure 9.2 Image courtesy of [Nießner et al. 13], ACM 2013

Figure 9.3 Image courtesy of [Keller et al. 13], IEEE 2013

Figure 13.2 Figures adapted from [de Aguiar et al. 08a]

Figure 13.5 Figures adapted from [Neumann et al. 13b]

Figure 15.1 Image courtesy of Michael Stengel

Figure 15.2 Image courtesy of Michael Stengel

Figure 15.3 Image courtesy of Michael Stengel

Figure 15.4 Image courtesy of Michael Stengel

Figure 15.5 Image courtesy of Mirko Sattler, Ralf Sarlette, and Reinhard Klein

Figure 15.6 Image courtesy of Michael Stengel

Figure 15.7 Image courtesy of Anna Hilsmann

Figure 17.1 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.2 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.3 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.4 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.5 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.6 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.7 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.8 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.9 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.10 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 17.11 3D Model "Iggy" by Dan Vulanovic was used for this image under the CC-Attribution 3.0 license

Figure 20.2 Image courtesy of Filmakademie Baden-Württemberg, The Gathering 2011

Figure 21.5 Images courtesy of Philippe Bekaert and Tom Mertens, Hasselt University, EU "2020 3D Media" project

Figure 21.6 Images courtesy of Philippe Bekaert at Hasselt University, Belgium, iMinds "explorative television" project

Figure 22.3 Image courtesy of [Schwartz et al. 11a], Eurographics Association 2013 / 2011

Figure 22.5 Image courtesy of [Lavoué et al. 13], ACM 2013

Figure 22.7 Image courtesy of [Schwartz et al. 11a], Eurographics Association 2011

Figure 23.1 Image courtesy of Filmakademie Baden-Württemberg, Jahre Leben 2013

Figure 23.3 Image courtesy of Filmakademie Baden-Württemberg, Dark Matter 2014

The Editors

Marcus Magnor is professor of computer science at Technische Universität (TU) Braunschweig, Germany, where he is chair of the computer graphics lab. He also holds an appointment as adjunct professor in the Physics and Astronomy Department at the University of New Mexico, USA. He earned his BA (1995) and MS (1997) in physics from Würzburg University and the University of New Mexico, respectively, and his PhD (2000) in electrical engineering from Erlangen University. After his postdoctoral time at Stanford University, he joined the Max-Planck-Institut Informatik in Saarbrücken as Independent Research Group leader. He completed his habilitation in 2005 and received the venia legendi for computer science from Saarland University. His research interests center around the natural phenomenon of images, from their formation, acquisition, and analysis to image synthesis, display, perception, and cognition. Areas of research include, but are not limited to, computer graphics, vision, visual perception, image processing, computational photography, astrophysics, imaging, optics, visual analytics, and visualization. He is the recipient of an ERC Starting Grant as well as being a Fulbright scholar, an elected member of the Braunschweigische Wissenschaftliche Gesellschaft, and laureate of the Wissenschaftspreis Niedersachsen.

Oliver Grau joined Intel as associate director of operations of the Intel Visual Computing Institute in Germany in October 2012. He earned a PhD from the University of Hannover, Germany, in 1999. Prior to Intel he worked for BBC R&D in the UK on innovative tools for visual media production. Since 2013 he has been a visiting professor at University of Surrey, UK. Oliver's research interests are in the intersection of computer vision and computer graphics techniques. His prior work included immersive virtual production systems, stereoscopic video production tools, free-viewpoint visualization of sport scenes, and Web-delivery of free-viewpoint experiences. More recent research interests include visual computing for new user experiences and digital content creation tools. Dr. Grau has a long track history of leading interdisciplinary work in more than 10 major

collaborative projects, between academic and industrial partners. He has published a number of scientific papers and holds several patents.

Olga Sorkine-Hornung is an associate professor of computer science at ETH Zurich, where she leads the interactive geometry lab at the Institute of Visual Computing. Prior to joining ETH she was an assistant professor at the Courant Institute of Mathematical Sciences, New York University (2008–2011). She earned her BSc in mathematics and computer science and PhD in computer science from Tel Aviv University (2000, 2006). Following her studies, she received the Alexander von Humboldt Foundation Fellowship and spent two years as a postdoc at the Technical University of Berlin. Professor Dr. Sorkine-Hornung is interested in theoretical foundations and practical algorithms for digital content creation tasks, such as shape representation and editing, artistic modeling techniques, computer animation, and digital image manipulation. She also works on fundamental problems in digital geometry processing, including reconstruction, parameterization, filtering, and compression of geometric data. Professor Dr. Sorkine-Hornung received the EUROGRAPHICS Young Researcher Award (2008), the ACM SIGGRAPH Significant New Researcher Award (2011), the ERC Starting Grant (2012), the ETH Latsis Prize (2012), and the Intel Early Career Faculty Award (2013).

Christian Theobalt is a professor of computer science and the head of the research group "Graphics, Vision, & Video" at the Max-Planck-Institute for Informatics, Saarbrücken, Germany. From 2007 until 2009 he was a visiting assistant professor at Stanford University. He earned his MSc degree in artificial intelligence from the University of Edinburgh, Scotland, and his Diplom (MS) degree in computer science from Saarland University, in 2000 and 2001, respectively. In 2005, he earned his PhD (Dr.-Ing.) from Saarland University and the Max Planck Institute for Informatics. His research lies on the boundary between computer vision and computer graphics. For instance, he works on 4D scene reconstruction, marker-less motion capture, machine learning for graphics and vision, and new sensors for 3D acquisition. Dr. Theobalt has received several awards: The Otto Hahn Medal of the Max–Planck Society (2007), the EUROGRAPHICS Young Researcher Award (2009), the German Pattern Recognition Award (2012), and an ERC Starting Grant (2013). He is also a co-founder of the Captury (www.thecaptury.com).

Acknowledgments

This book is the result of work by experts from the fields of computer graphics, computer vision, and visual media production. We are deeply indebted to all contributing authors and thank everyone for the considerable time and effort they have devoted to this project. The idea for this book came about in the fall of 2013 at the Dagstuhl seminar on Real-World Visual Computing. Schloss Dagstuhl, the Leibniz Center for Informatics, situated in the peaceful and picturesquely forested hills of the northern Saarland in the westernmost part of Germany, offers computer scientists from all over the world the unique opportunity to get together for a full week to present their latest research, discuss and exchange novel ideas, and to get to know each other on a personal level. We thank the staff of Schloss Dagstuhl for their heartwarming hospitality as well as for the superb cuisine that kept everyone's body, soul, and mind together (or as the Saarland natives say: "Hauptsach gudd gess").

The content presented in this book constitutes mostly fundamental research that is available for everyone to read, re-implement, and use for their own purposes, free of charge. This is possible only because of publicly funded research. We gratefully acknowledge the support from all the funding agencies who invested in the research the results of which are presented in this book, in particular the German Science Foundation (DFG), the Swiss National Science Foundation (SNF), and the European Research Council (ERC).

We thank all the people at CRC Press who have helped us in getting this book written, edited, proofread, printed, and published in such a short time. We would explicitly like to thank Sarah Chow and Joselyn Banks-Kyle for their great help and support.

While all of the above were necessary ingredients to make this book happen, there is one person without whom the book would not have come into existence. Felix Klose was the good soul of our project. He prepared the LaTeX templates, set up the project's Wiki pages, reminded authors

of deadlines, collected permission forms, made sure all chapter files were compiled, and much more. Felix, thank you for your commitment and perseverance!

The Editors
Marcus Magnor, Oliver Grau, Olga Sorkine-Hornung, Christian Theobalt

Part I

Acquiring the Real World

1
Camera Sensor Pipeline

Jan Kautz, Hendrik P.A. Lensch, Céline Loscos, and Philippe Bekaert

1.1 Introduction

The very first step of most real-world visual computing applications is the acquisition of images or video. However, acquiring meaningful image and video data is surprisingly challenging. This stems from the fact that real-world cameras and sensors are far from perfect concerning sampling or measuring and a substantial amount of processing needs to be applied before the data can be used. The different sensor types will be discussed, in particular with regard to how they affect the quality of data, how measurement noise affects image acquisition, and how to create dense color samples from sparse samples as they are acquired by most cameras. In order to characterize a given camera, it has to be calibrated in terms of radiometry and color as well as lens and geometric calibration. Applying all these steps results in well-calibrated and meaningful images.

1.2 Sensor Technology

Most digital imaging sensors operate based on the inner photoelectric effect. In the depletion area of the p-n junction of a photo diode an incoming photon of sufficient energy will create an electron-hole pair producing a photo current. In principle, every photon of sufficient energy (wavelength) can contribute to the effect, but the specific *quantum efficiency* depends on the wavelength. For a silicon photo diode the response covers the visible and the near infrared spectrum (400-1000nm).

The photoelectric effect produces a current that is linearly proportional to the radiant power, i.e., it can be used for physical measurements for all practical considerations inside a camera. Only at very strong illumination non-linear effects might occur [Anisimov et al. 77].

The light sensitive part of a pixel on a sensor typically is a photo diode. Besides photo diodes cameras might otherwise employ photo transistors

with the added benefit of preventing further exposure by electronically closing the gate.

The charge collected during exposure needs to be stored, amplified and finally converted to a digital value. The design of a sensor allocates resources for these steps. The well capacity indicates how many electrons can be accumulated in one exposure, the scale indicates how much the photo current is amplified, and the bit rate is correlated to the number of discernible intensity values.

There are two fundamentally different approaches on how the charge is transfered to the A/D unit. In charge coupled devices (CCD) the charge is transported pixel by pixel to the end of each row, and the pixels in the last row are successively piped through a single amplifier and converter. Transfer between pixels can be carried out with hardly any loss. The main benefit of a CCD is that the information gathered by all pixels is basically processed by the same amplifier and converter. They undergo the same transformation. As a draw-back, a CCD can only read out rectangular regions. The speed of a CCD is also limited by the frequency of the A/D unit. As quality is typically decreasing with speed, reading off a multi-megapixel CCD at highest quality can take up to a couple of seconds. The process can be accelerated by providing multiple A/D units and splitting the sensor plane into tabs. This, on the other hand, leads to difficult to control conversion settings which are independent for each tab. In video cameras often interlaced read-out is used to provide higher frame rate. Each frame contains only half the rows, specifically every other row. Two subsequent frames alternate between the two sets of rows. The process of de-interlacing then performs a spatio-temporal interpolation between these to half-frames.

The second approach often employed is based on CMOS technology with the ability to group more electronic processing close to each pixel. Similar to random access memory, pixels can be addressed per row or individually. From just reading off a few pixels one can for example quickly sample an image histogram. Each pixel is equipped with a small amplifier which in the early days led to rather noisy CMOS images as each pixel is amplified individually. The additional electronics per pixel reduces the space available for the photo-sensitive part, lowering the fill factor. A lens on top of each pixel can counteract this loss in fill factor. Benefits of CMOS sensors are the flexibility of addressing and lower production cost.

More exotic sensors include for example back-illuminated CCDs where the support structure is thinned and the illumination is provided from the back-side avoiding photons being blocked by the electronic wires. For light sensitive applications this approach is typically combined with electron multiplying CCD (EMCCD) that employ solid-state impact ionization to multiply the number of generated photo-electrons. On the CMOS side

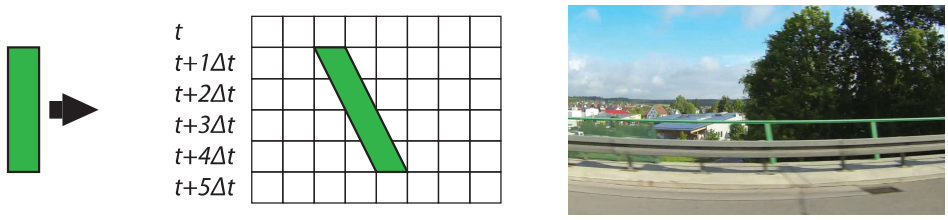


Figure 1.1: Rolling Shutter. In a rolling-shutter sensor each row will start exposure at a slightly different time. This results in distortion of moving parts. A vertical line moved to the right will be sheared.

so-called scientific CMOS sensors provide high-quality imaging by a more elaborate design of the per-pixel amplifiers.

Another important factor on sensors is how the entire image is read off the chip. Some CCD sensors provide a shielded area for storing the accumulated charge of one exposure while the sensor still is illuminated. As this electronic shutter transfer is synchronous one obtains the same global shutter for all pixels.

This is in contrast to the cheaper and faster rolling shutter most often found in CMOS chips where some rows are read out while others are still being exposed (Figure 1.1). In order to guarantee the same exposure duration, the exposure and read out of the rows is staggered. As a consequence the different rows will capture the scene at different moments in time. Special care is necessary when employing rolling shutter cameras for 3D reconstruction in dynamic environments as each camera (each row) potentially captures a different slice of the space time volume. Even though two cameras expose synchronously the same scene feature might be recorded at different times depending on its position in the respective camera image.

1.3 Noise

Inherent to digital imaging are a number of noise sources that affect every captured image. Reibel et al. [Reibel et al. 03] discerns two major classes: *temporal* and *non-temporal* noise.

The temporal noise sources vary with the scene brightness, and the temperature of the sensor. A fundamental limit to the accuracy of photographic measurements is the photon shot noise. Any source of light creates photons according to a temporal Poisson random process, i.e., the rate at which photons arrive at the sensor fluctuates. The variance of the photon shot noise is linearly correlated to the light intensity. Therefore, the standard deviation and at the same time the signal-to-noise ratio increases

with the square root of the signal (SNR $= N/\sqrt{(N)} = \sqrt{(N)}$ for N photons). Similarly, heat can knock electrons loose in the silicon, producing a so-called dark current. The effect is independent of the actual signal, but the dark current can limit the maximum exposure duration when exceeding the well capacity. Dark current enters the subsequent amplification and A/D conversion step. Thus, these electrons are indistinguishable from photo electrons. Cooling the camera reduces the effect of the dark current shot noise as the noise doubles every $5 - 8°$ Celsius. Another temporal source of noise is the amplification and conversion step where thermal noise and a frequency-dependent flicker noise in the amplifier as well as quantization in the digitization step degrade the signal.

Non-temporal noise occurs due to static defects of the sensor. Due to slight irregularities, the area of the photo-sensitive part might vary and the properties of the per-pixel electronics might differ. The amount of dark current varies from pixel to pixel, resulting in a fixed pattern per-pixel bias independent of the signal. Similarly, the effect of photo-response non-uniformity corresponds to the amplifier gain being different per pixel. Some pixels reach saturation earlier than others, a problem mainly found in CMOS sensors. Finally, the actual amplification might not be perfectly linear, corrupting the direct linear relationship between photons and electrons. For HDR imaging, therefore, the actual photon transfer curve needs to be estimated (see Section 1.5).

The individual noise sources co-occur all at the same time during image capture and cannot always be disentangled. If accurate photometric calibration is required, cooling and taking a number of calibration images can improve image quality and allows to quantify the potential variance [Granados et al. 10, Hasinoff et al. 10]. Most common is to capture and average a series of dark frames with the same exposure time as the intended shot but leaving the cover on the lens. This way, the dark current and its spatial non-uniformity can be characterized. The variance of the readout noise can be captured by a bias frame, an image of zero exposure time. In order to quantify the photo-response non-uniformity, i.e., the per-pixel bias, a flat field is needed, a picture taken without a lens where each pixel receives exactly the same exposure. A practical difficulty is to ensure a really uniform illumination on the sensor. Perfect would be a large homogeneous area light source such as a monitor with added diffusor or a quite distant point light source. In a similar way a flat field captured with the lens can correct for vignetting. Considering all these measures, Granados et al. [Granados et al. 10] developed a noise-optimal pipeline for combining multi-exposure photos into a single HDR image.

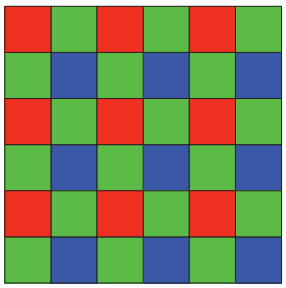


Figure 1.2: Example of the common Bayer color filter array (CFA).

1.4 Demosaicing and Noise Reduction

Most digital cameras are single sensor cameras, i.e., only a single sensor measures the incoming light. However, a pixel in a CCD or CMOS sensor cannot sense the wavelength of the incoming light, but only its power. To enable color imaging, a color filter array (CFA) is overlaid on the sensor pixels: each pixel now senses only light within a specific wavelength range, typically corresponding to red, green, and blue wavelengths. The most common pattern is the Bayer pattern, with one red pixel, two green pixels, and one blue pixel in each 2×2 block of pixels (Figure 1.2). The use of a CFA leads to colors being sensed sparsely and missing color information needs to be filled in. This process is commonly called demosaicing, and many different techniques have been proposed over the years [Li et al. 08].

The simplest method is to simply take all the samples for a given color channel and to bilinearly interpolate from the nearest neighbors [Longere et al. 02]. As one might expect, this yields artifacts across edges and in areas with high-frequency texture content, since correlation between color channels is not taken into account. For instance, if there is a strong discontinuity between two neighboring green pixels, there is a high chance that there is a discontinuity also in the red and blue channels, but simple per-channel bilinear interpolation cannot reproduce this.

Quality can be increased with gradient-based methods, which typically estimate a local gradient direction followed by filtering along estimated edge directions and not across, thus avoiding the issues discussed above. The well-known Malvar–He–Cutler demosaicer (the default method in MATLAB®) falls into this category [Malvar et al. 04]. It still performs bilinear interpolation, but corrects it with a local gradient estimate using a 5×5 pixel window. This yields much improved results but can still lead to "zippering" artifacts, i.e., a visible high-frequency pixel pattern along high-frequency edges.

The best quality can be achieved by exploiting image self-similarity [Zhang et al. 11]. Instead of trying to estimate local image features across sparsely sampled color channels, self-similarity is used to derive the missing information. The LDI-NLM algorithm (and the very similar LDI-NAT), works as follows [Zhang et al. 11]. First, a standard directional interpolation method as described above is used to create an initial estimate of the green-channel. The green channel is then enhanced by running non-local means (NLM) [Buades et al. 05] on it. NLM will find similar patches for each pixel and compute a weighted average of those patches, which in turn is likely to improve the interpolated samples as additional data is being used. Following the reconstruction of the green channel, an initial estimate of the R and B channels are created (using information from the now complete green channel). Then, NLM is again run on the initial red and green channels. The LDI-NAT version proceeds similarly but uses soft thresholding in a sparse transform domain (similar to the BM3D denoising algorithm [Dabov et al. 07]). LDI-NLM and LDI-NAT achieve excellent results and outperform most other methods.

Noise Reduction It is important to note that these demosaicing methods assume noise-free input data. Of course, this is not usually the case. Applying these methods to noisy input data, however, often emphasizes color noise. Subsequent denoising (e.g., using the state-of-the-art BM3D denoiser [Dabov et al. 07]) of the demosaiced images is then necessary. Joint demosaicing and denoising is possible, but only little research has been conducted in this area to date [Chatterjee et al. 11].

1.5 Radiometry and Color

Sensing Radiance

As described in Section 1.2, the A/D unit converts the charge of each pixel to a digital value. This conversion is directly proportional to the charge, i.e., linear in the number of photoelectrons that have reached the sensor pixel (discounting noise). Most professional cameras allow the user to access this raw data, i.e., without any post-processing such as white-balancing, gamma correction, noise reduction, and so forth. If the raw data cannot be accessed on a particular camera, it is still possible to calibrate the response curve of the camera.

Color

Different sensors use different color filter arrays and different manufacturing processes, which leads to device-dependent color measurements. To output

physically meaningful and device-independent color coordinates, such as CIEXYZ or CIELAB, the camera must be calibrated. This process is often called device characterization and requires two components [Johnson 02]: 1) calibration: determining the device's color space; and 2) characterization: finding a mapping between the device color space and the device-independent color space, e.g., CIE tristimulus values.

Suppose a color target is being captured. In discretized form, the trichromatic response value $[R, G, B]$ of a specific pixel on the sensor is given as the sum of the product of the spectral power distribution (irradiance) of the light source $P(\lambda)$, the surface reflectance of the imaged object $S(\lambda)$, and the spectral sensitivities of the color filters $D_{r/g/b}(\lambda)$:

$$R = \sum_{\lambda} P(\lambda)S(\lambda)D_r(\lambda)\Delta\lambda, \tag{1.1}$$

$$G = \sum_{\lambda} P(\lambda)S(\lambda)D_g(\lambda)\Delta\lambda, \tag{1.2}$$

$$B = \sum_{\lambda} P(\lambda)S(\lambda)D_b(\lambda)\Delta\lambda, \tag{1.3}$$

where the summation is over the visible spectrum. Now this is very similar to the computation of device-independent color values, such as CIEXYZ:

$$X = \sum_{\lambda} P(\lambda)S(\lambda)\bar{x}(\lambda)\Delta\lambda, \tag{1.4}$$

$$Y = \sum_{\lambda} P(\lambda)S(\lambda)\bar{y}(\lambda)\Delta\lambda, \tag{1.5}$$

$$Z = \sum_{\lambda} P(\lambda)S(\lambda)\bar{z}(\lambda)\Delta\lambda, \tag{1.6}$$

where $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ are the CIE color matching functions. So the only difference is the device-dependent color $D_{r,g,b}$ vs. the device-independent functions $\bar{x}, \bar{y}, \bar{z}$.

Many characterization techniques have been proposed. They largely fall into two categories: reflectance-based characterization, and characterization based on monochromatic light. Reflectance-based characterization usually requires a color target with known reflectances and a suitable sampling of the color space, such as a GretagMacbeth ColorChecker, of which a picture is taken. A direct mapping between the (raw) images RGB-values and the known XYZ values of the color target can be derived via linear regression. While these techniques are very easy to use, they are only valid for the current illumination condition as the illuminant $P(\lambda)$ is "baked in." The most common monochromator-based method uses a hollow white sphere, which is illuminated by a monochromator with an adjustable wavelength.

An image is taken for a number of wavelengths, which allows for a direct mapping between the device's color coordinates and CIEXYZ tristimulus values. While this is a time-consuming and expensive calibration method, it is vertically accurate.

CIEXYZ is the basis from which one can convert to many other common color spaces, such as sRGB. sRGB is notable because it has found widespread use, as it was designed for typical home viewing conditions and not darker environments that are used by professionals for color matching. It is a non-linear color space, with an overall gamma of about 2.2 but consisting of a linear plus a non-linear part.

HDR Imaging

High-dynamic range (HDR) imaging allows for the representation of a larger range of intensities than conventional images [Reinhard et al. 08, Reinhard et al. 10]. It is widely used by photographers and supported by software[1] to avoid saturated areas or under-exposed pixels. It is also used to acquire more precise illuminant information of the real world when modeling objects, or to guide image compositing for coherent common illumination when mixing real and virtual content.

Conventional camera sensors typically digitize luminance with 8 to 16 bits. Even when digitized with 16-bit accuracy natural scenes can still easily exceed the dynamic range of the sensor. There are many definitions of what is high-dynamic range. Some consider that non-linearly representing the range of luminance using 8 bits qualifies for HDR. Others consider HDR to be the full variation of the physical luminance of the real world that the human visual system is capable of adapting to, thus 10 orders of magnitude. Recently, a group of experts[2] came to the consensus that high-dynamic range should represent the perceptual range of intensities simultaneously perceivable by the human eye, thus 6 orders of magnitude, which can be stored on a 20-bit image.

There exist two main procedures to capture HDR content: by merging conventional camera images, or by providing enhanced hardware capability. In order to create an HDR image with a conventional camera, images are taken with different time exposures in order to capture different ranges of luminance. Combining these images requires two steps: radiometric calibration and merging values into HDR data. Radiometric calibration is necessary mostly if RAW sensor data are not available or very noisy. It consists of finding a linear color mapping from one image to another that are taken with different exposures. Merging values into HDR data consists of carefully selecting pixels from all images to form a coherent HDR image.

[1] e.g., Adobe Photoshop - http://www.adobe.com/fr/products/photoshop.html
[2] HDRi - COST Action IC1005 - http://www.ic1005-hdri.com/

Enhancing hardware capabilities corresponds to increasing the dynamic range of sensors. SpheronVR,[3] for example, provides cameras (photographic and video) with sensors capable pf covering 8 orders of magnitude. These cameras are not aimed at the general public, and some of them are at the stage of prototypes, limited by streaming and storage facilities. Other technologies involve using beam splitters [Aggarwal and Ahuja 04, Tocci et al. 11] to capture data at different intensities with a single camera and a single shot. Merging is done in a similar way as for sequential multi-exposure images. Finally, it is possible to adapt a mask in front of the sensing array with a pattern to reduce the incoming light to different degrees, and to produce spatially varying exposures [Nayar and Mitsunaga 00]. Beam splitter-based approaches as well as spatially varying exposure approaches provide the advantage that they can be directly applied to dynamic, time varying scenes since all images represent the same instant. These types of approaches, though, are limited in the captured range by their beam splitter capability and the spatially varying exposures respectively.

Radiometric Calibration Displays and cameras employ a response function to modify measured luminance to create pleasant overall colors when perceived by the human eye. For color image processing, radiometric calibration needs to be performed. In the case of high-dynamic range images, we need to find the inverse response function of the camera to linearize pixel color relations. Ideally, inter-image relation should lead to the radiometric relation for a 3D point that projects to the same image coordinates (x, y) of two images $\mathbf{I}^0$ taken at exposure time t_0 and $\mathbf{I}^1$ taken at exposure time t_1, linking the radiance E arriving at sensors and stored in images as RGB values:

$$E_{I_0}/t_0 = E_{I_1}/t_1 \tag{1.7}$$

RAW sensor information can be used directly with this equation to transform pixel color values to coherent radiance values in all images. However, depending on the camera, this is not always true, and even more when no access to RAW data is possible. There is a need to find the inverse camera function $g = f^{-1}$, with f non-linearly transforming the radiance values to color. Inverting the function is possible because values monotonically increase. Several methods have been proposed [Mann and Picard 95, Mitsunaga and Nayar 99, Grossberg and Nayar 04, Debevec and Malik 97]. They all fit a curve to selected values and therefore are approximative. However, this is generally sufficient, and remaining small errors can be compensated in the HDR reconstruction phase.

[3] https://www.spheron.com/home.html

HDR Reconstruction HDR Reconstruction is the process of merging values coming from different images into one coherent HDR value. The general equation for N images and the pixel colors $E_i(x, y)$ of each image i at coordinates (x, y) is:

$$E(x,y) = \frac{\sum_{i=0}^{N} \omega(I_{p_i}) \frac{g(E_i(x,y))}{t_i}}{\sum_{i=0}^{N} \omega(I_{p_i})} \tag{1.8}$$

The difficulty here is to chose the weights $\omega(I_{p_i})$ associated with the pixel I_{p_i} of image i. They are used to enhance or reduce the impact of pixel colors in the final HDR result [Granados et al. 10]. The weight function excludes under- and over-saturated pixels [Debevec and Malik 97] but can also be based on signal-to-noise ratio [Mitsunaga and Nayar 99].

This reconstruction approach assumes that images are perfectly aligned and that no movement occurred during sequence acquisition. If this is not the case, weights can also reflect the probability of a pixel to belong to the background [Khan et al. 06]. For motion registration or non-aligned cameras, more complicated operations need to be performed to register pixels before reconstruction can be achieved [Loscos and Jacobs 10, Bonnard et al. 13].

Multispectral Imaging

The quantum efficiency of silicon-based camera sensors is by itself a wavelength-dependent function. The Foveon sensor was able to detect color by measuring at three different penetration depths in the silicon. However, this concept has never been extended to more than three wavelength bands. The most common approaches for capturing more than three color channels are either to extend the Bayer pattern and include more colors, or to use a second optically aligned sensor with a Bayer pattern of different base colors.

If significantly more wavelength bands are required, there are basically two different approaches:

The first approach captures one wavelength band at a time using a filter wheel or a tunable filter. Tunable filters employ an electro-optic or acusto-optic effect to transmit only the selected wavelength band. The drawback of this filtering approach is that only a small fraction of the overall radiant power is captured in each band, resulting in a lengthy process to capture a multispectral image.

The second approach makes use of a prism or diffraction grating to split up the incoming light into its spectrum. Once spatially separated, the different wavelengths can be modulated individually and then recombined onto the sensor [Mohan et al. 08, Kim et al. 12]. The benefit is that the

entire spectrum can be varied, although not necessarily in the same way for the entire image plane, rather than selecting only a single wavelength band. In order to produce a multi-channel image, this optical setup is often combined with compressed sensing approaches [Mohan et al. 08, Kim et al. 12].

1.6 Geometric Calibration

Applications such as 3D geometry reconstruction, view interpolation, and so on require understanding the mapping between 3D real scene points and image coordinates. The process of determining the actual value of the parameters that control that mapping is called geometric camera calibration. In this section, a common model for this mapping is presented, and the basic principles of lens distortion, intrinsic and extrinsic single-camera calibration are outlined. The simultaneous calibration of multiple cameras is explained in Section 2.3.

Camera Calibration Parameters

The geometric camera calibration parameters fall into four categories: sensor-related parameters, lens-related parameters, camera-lens assembly parameters, and extrinsic parameters.

Sensor-related paramaters include the *image width and height* in pixels, and the *pixel pitch*: the spacing of pixels in each row and between rows. They are usually known from camera specifications and region of interest settings.

Lens-related parameters do not depend on the camera the lens is mounted on, nor its position and orientation in space. They include the lens image formation model and lens distortion. Most lenses are rectilinear lenses, ideally mapping straight world lines to straight image lines. They are characterized by their *focal length*. Equidistant fish eye ($f\theta$) lenses can offer greater sharpness and less distortion for wide viewing angles. Also these lenses are characterized by their focal length f, which has however a different meaning than for rectilinear lenses. *Lens distortion* models quantify the deviation of a real lens from the ideal rectilinear or $f\theta$ model.

Camera-lens assembly parameters include the *principal point*, the *center of distortion* and *effective pixel aspect ratio and skew angle*. The principal point is the image coordinate of the intersection of the optical symmetry axis of a lens with the camera sensor plane. The center of distortion usually is equal to the principal point. The effective pixel aspect ratio and skew angle may deviate slightly from sensor specifications due to mechanical tolerances in lens and camera housing.

Table 1.1: Set of geometric camera calibration parameters.

symbol	parameter name	unit
w,h	pixel width and height	micrometers
$k_0, k_1, \ldots$	lens distortion coefficients	$1/cm^\kappa$
x_d	center of distortion	image coordinates
f	focal length	millimeters
x_c	principal point	image coordinates
a	effective aspect ratio	dimensionless
θ_{skew}	effective skew angle	degrees
α, β, γ	camera orientation Euler angles	degrees
C	optical center	world coordinates

The *position and orientation* of a camera in 3D real-world space are called the *extrinsic* parameters of the camera. Position is always relative to a particular choice of 3D real world coordinate system. The position that counts is the *optical center*: the point in 3D space where rays of light hiitting the lens would meet, if they were not bent to focus on the sensor. For humans, orientation is conventiently expressed by means of Euler angles (note there are 24 different interpretations of Euler angles [Schoemake 94]). In computations, quaternions, exponential maps, or a rotation matrix will usually be preferred.

Table 1.1 summarizes a typical set of geometric camera calibration parameters.

Mapping World Space Points to Image Coordinates

Mapping world space point X to image coordinates x basically takes four steps:

- mapping world space point X to camera space point X_c;
- application of the image formation model to map X_c to a location x_f on the lens focal plane, relative to the principal point;
- mapping lens focal plane position x_f to an ideal (undistorted) image coordinate $\bar{\mathrm{x}}$;
- applying the lens distortion model to obtain the observable (distorted) image coordinate x.

Lens distortion is part of image formation by the lens in physical reality. In visual computing, however, it is usually modelled as a correction to ideal image coordinates as described here.

Mapping image coordinates to world space rays takes the inverse of these steps, applied in reverse order.

The first step, is a simple translation taking the world origin to the cameras optical center C, and rotation $\mathbf{R}$ aligning the view to a canonical axis system, such as in OpenGL (view direction is negative Z, image right direction is X, image up direction is Y):

$$\mathrm{X}_c = \mathbf{M}\mathrm{X} \qquad \text{with} \qquad \mathbf{M} = \left[\mathbf{R}^\top \middle| -\mathbf{R}^\top \mathrm{C}\right]. \tag{1.9}$$

The matrix $\mathbf{M}$ is called the camera *extrinsic matrix*.

For rectilinear lenses, the second step is a rescaling of X and Y by inverse depth $-1/Z$ and focal length f:

$$x_r = f\frac{X}{-Z} \quad y_r = f\frac{Y}{-Z} \quad r = f\tan\theta.$$

θ is the angle between the optical axis of the lens and the incident light ray direction. r is the distance in millimeters (if f is expressed in millimeters) of the light ray projection on the focus plane, relative to the principal point. For other lens models, other similar formulae apply (such as $r = f\theta$ for equidistant fish eye lenses). The minus sign is due to our coordinate system convention (OpenGL-style $Z < 0$ in front of the camera).

For a rectangular sensor pixel grid, and in absence of distortions causing pixel grid skew or aspect ratio abberations, the third step is a simple 2D scaling and translation from sensor plane position in millimeters relative to the principal point to pixel unit distance with respect to the top-left image corner or other chosen image coordinate origin. In general, it is a 2D shearing transform taking into account effective aspect ratio and skew angle.

For rectilinear lenses, steps two and three can be combined into a single matrix multiplication, yielding *homogeneous* undistorted image coordinates. These require *perspective division* of $\bar{x}$ and $\bar{y}$ by $\bar{z}$ in order to obtain affine image coordinates (Table 1.1):

$$\begin{bmatrix}\bar{x}\\ \bar{y}\\ \bar{z}\end{bmatrix} = \mathbf{K}\begin{bmatrix}X_c\\ Y_c\\ Z_c\end{bmatrix} \quad \text{with} \quad \mathbf{K} = \begin{bmatrix}\tilde{f} & -\tilde{f}\tilde{s} & -x_c\\ 0 & -\tilde{f}\tilde{a} & -y_c\\ 0 & 0 & -1\end{bmatrix} \tag{1.10}$$
$$\tilde{f} = f/w \quad , \quad \tilde{s} = -\tan\theta_{skew}. \quad , \quad \tilde{a} = a/\cos\theta_{skew}$$

This matrix $\mathbf{K}$ is named the *intrinsic camera matrix*. The minus signs in the definition of $\mathbf{K}$ are due to our coordinate system conventions (OpenGL style $Z < 0$ in front of the camera, Y pointing up, image y pointing down given image coordinate origin in the top-left corner).

For rectilinear lenses, the full mapping from homogeneous world coordinates to homogeneous undistorted image coordinates can be obtained as

a single matrix-vector product:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad \text{with} \quad \mathbf{P} = \mathbf{KM} \tag{1.11}$$

The matrix $\mathbf{P}$ is called the *full camera matrix*.

A most common model for lens distortion, the fourth step, is the following [Brown 66, Slama 80, Heikkila and Silven 97, Zhang 00]:

$$\begin{aligned} \mathrm{x} &= \mathrm{x}_d + L(\bar{\mathrm{x}}') \quad , \quad L(\bar{\mathrm{x}}') = \bar{\mathrm{x}}' L_r(r) + L_t(\bar{\mathrm{x}}') \\ \bar{\mathrm{x}}' &= \bar{\mathrm{x}} - \mathrm{x}_d \quad , \quad r = \sqrt{\bar{x}'^2 + \bar{y}'^2} \\ L_r(r) &= k_0 + k_1 r^2 + k_2 r^4 + k_3 r^6 \\ L_t(x,y) &= (p_1 B(x,y) + p_2 D(x,y), p_2 C(x,y) + p_1 D(x,y)) \\ B(x,y) &= 3x^2 + y^2 \quad , \quad C(x,y) = x^2 + 3y^2 \quad , \quad D(x,y) = 2xy \end{aligned} \quad . \tag{1.12}$$

The model consists of a radial part L_r, modifying distance with respect to the *center of distortion* x_d, and a tangential part L_t.

Lens Distortion Calibration

Lens distortion is calibrated when a set of lens distortion parameter values has been found that warps a captured image into an image that shows straight world lines as straight image lines. The distortion parameters are the distortion coefficients k_i, as well as the center of distortion (x_d, y_d).

Auto-Calibration In order to calibrate lens distortion under uncontrolled circumstances, one or a few images of scenery exhibiting straight world lines suffices, such as windows or doors in an image of a building facade. Lens distortion parameters can be obtained by non-linear optimization, e.g., with the Levenberg–Marquardt algorithm. In each step of the optimization procedure, edge pixel locations in the input image(s) are warped using the (inverse) lens distortion model. The quality of the parameter set is evaluated by measuring to what extent the warped edge pixels form straight lines [Devernay and Faugeras 01]. A practical tool implementing a similar approach, is PTLens.[4]

Lens Distortion from Calibration Grids Often in stereo- or multi-view setups, lens distortion can be calibrated in controlled lab circumstances. Known patterns of features are filmed and analyzed. Often used patterns include

[4] http://epaperpress.com/ptlens/

planar checkerboard calibration patterns (using saddle-points) and rectangular grids of circular dots (using centroids).

In absense of lens distortion, the relation between the known 2D real world calibration grid feature positions and their image coordinates, is a planar perspective transform, also called a *2D homography* [Hartley and Zisserman 03, §2.3]. Distortion parameters can be estimated by iterative optimization algorithms that minimize the deviation of correspondences from a 2D homography. The distortion center can also be estimated using direct techniques [Hartley and Kang 05]. More direct estimation techniques, based on *lifted coordinates* are described in [Sturm et al. 11]. These techniques can be generalized to non-rectilinear lenses.

Intrinsic Calibration

Estimating the intrinsic camera calibration parameters is the determination of camera parameters determining the mapping between (ideal, undistorted) image coordinates and camera space ray directions. For rectilinear cameras, this mapping is governed by the intrinsic camera matrix K (Equation 1.10).

From camera specifications, pixel aspect ratio a and skew angle θ_{skew} are typically known to sufficient accuracy. Often, pixels are square. When a camera is equipped with a lens exhibiting lens distortion, the principal point x_c may be taken equal to the lens distortion center x_d calculated using above sketched methods. The main intrinsic parameters to be determined thus typically are the principal point $\mathrm{x_c} = (x_c, y_c)$ for a lens without significant distortion, and the focal length f.

Instrinsic camera parameters can be auto-calibrated from observations of orthogonal world lines and/or planes, or determined from 2D homographies relating a planar calibration grid with its image taken at different angles [Zhang 00].

These observations impose linear *constraints* on a particular symmetric 3×3 matrix $\omega = \left(\mathbf{K}\mathbf{K}^\top\right)^{-1}$, called the *image of the absolute conic (IAC)*. Consider, for instance, the vanishing points v_1 and v_2 of two (bundles of) orthogonal lines with direction vectors $\mathrm{D}_1 = (X_1, Y_1, Z_1, 0)$ and $\mathrm{D}_2 = (X_2, Y_2, Z_2, 0)$. Since the homogeneous component is zero, the relation between vanishing point v and affine direction vector $\mathrm{d} = (X, Y, Z)$, is:

$$\mathrm{v} = \mathbf{P}\mathrm{D} = \mathbf{K}\left[\mathbf{R}^\top \middle| -\mathbf{R}^\top \mathrm{C}\right] \begin{bmatrix} X \\ Y \\ Z \\ 0 \end{bmatrix} = \mathbf{K}\mathbf{R}^\top \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \Leftrightarrow \quad \mathrm{d} = \mathbf{R}\mathbf{K}^{-1}\mathrm{v}. \tag{1.13}$$

Since the direction vectors d_1 and d_2 are orthogonal,

$$0 = {\mathrm{d}_1}^\top \mathrm{d}_2 = {\mathrm{v}_1}^\top \mathbf{K}^{-\top}\mathbf{R}^\top\mathbf{R}\mathbf{K}^{-1}\mathrm{v}_2 = {\mathrm{v}_1}^\top \mathbf{K}^{-\top}\mathbf{K}^{-1}\mathrm{v}_2 = {\mathrm{v}_1}^\top \omega \mathrm{v}_2. \quad (1.14)$$

Such constraints on the IAC ω are stacked together into a homogenous linear system. This linear system is solved using singular value decomposition (SVD), with proper preconditioning. The thus found IAC is decomposed as $\omega = UU^\top$, U being an upper-triangular matrix, using Cholesky factorization. K is obtained as U^{-1}, and then decomposed into f, x_c, a and θ_{skew}, if required, and refined using Levenberg–Marquardt iterative optimization [Hartley and Zisserman 03, §8.6].

Extrinsic Calibration

Extrinsic calibration is the process of determining the location C and orientation $\mathbf{R}$ of a camera, or a set of cameras, with respect to a 3D world space coordinate system of choice.

The most straightforward way to find the location of fixed cameras is to measure them with a simple ruler or other distance measuring device. However, the exact location that matters is the optical center, which is the imaginary point in 3D space where the rays of light hitting the lens would meet if they were not bent to focus on the sensor. Its position relative to the camera body can be estimated typically only up to a few-centimeter precision.

Sometimes, location and/or orientation *tracker* devices, are used to measure camera positions and orientations. These devices can be based on mechanical, electrical, optical, magnetic, micro-electromechanical (MEM), electro-magnetic (EM, radio waves), or other principles [Danette Allen et al. 01]. GPS allows outdoor localization to an accuracy of about 1 meter, and update rate of one second typically. Compasses measure absolute orientation with respect to the earth magnetic field. For indoor use, optical tracking systems are often used, and regularly in combination with inertial tracking (with MEM devices). In all cases, it pays off to combine such measurements with visual tracking (Section 2.3).

Planar Calibration Grids The orientation $\mathbf{R}$ and location C of the camera, relative to a planar calibration grid, are easily obtained from a 2D homography $\mathbf{H}$, relating the grid with its image, and the intrinsic matrix $\mathbf{K}$.

Assume the calibration grid is in the world XY-plane ($Z = 0$). Let $\mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, \mathrm{p}_4$ denote the columns of the full camera matrix $\mathbf{P}$. Image points x are related with their corresponding calibration grid points

$\mathrm{X} = (X, Y, 0, 1)$, as follows:

$$\mathrm{x} = \mathbf{P}\mathrm{X} = \begin{bmatrix} \mathrm{p}_1 & \mathrm{p}_2 & \mathrm{p}_3 & \mathrm{p}_4 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{p}_1 & \mathrm{p}_2 & \mathrm{p}_4 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Therefore, $\mathbf{H} \doteq \begin{bmatrix} \mathrm{p}_1 & \mathrm{p}_2 & \mathrm{p}_4 \end{bmatrix}$, and since $\mathbf{P} = \mathbf{K} \begin{bmatrix} \mathbf{R}^\top \mid -\mathbf{R}^\top \mathrm{C} \end{bmatrix}$:

$$\mathbf{H} \doteq \mathbf{K} \begin{bmatrix} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{t} \end{bmatrix}. \tag{1.15}$$

Apart from sign and normalization, the first two columns r_1 and r_2 of $\mathbf{K}^{-1}\mathbf{H}$ provide the first two rows of $\mathbf{R}$. The third row is the cross product $\mathrm{r}_1 \times \mathrm{r}_2$. The optical center follows as $\mathrm{C} = -\mathbf{R}\mathrm{t}$. The fourfold ambiguity is resolved by testing each possible solution against the actual data.

This method is used in the popular camera calibration approach by Zhang [Zhang 00] and implemented in the camera calibration toolbox of Bouguet,[5] which is available in MATLAB and OpenCV.

3D Ground Control Points There is no straightforward way to estimate pose (position and orientation) of a camera relative to a set of world space points with known coordinates from their image projections. The full camera matrix P, however, can be estimated from 3D-2D correspondences as follows. When intrinsics are known, pose then can be obtained after full matrix estimation, as $\mathbf{M} = \mathbf{K}^{-1}\mathbf{P}$.

$$\mathrm{x} \doteq \mathbf{P}\mathrm{X} \quad \Longleftrightarrow \quad \begin{cases} kx = \mathrm{p}^1\mathrm{X} \\ ky = \mathrm{p}^2\mathrm{X} \\ kw = \mathrm{p}^3\mathrm{X} \end{cases} \quad \Longrightarrow \quad \begin{cases} \left(w\mathrm{p}^1 - x\mathrm{p}^3\right)\mathrm{X} = 0 \\ \left(w\mathrm{p}^2 - y\mathrm{p}^3\right)\mathrm{X} = 0 \end{cases}$$

k makes the scale ambiguity in $\mathrm{x} \doteq \mathbf{P}\mathrm{X}$ explicit. p^i denotes the i-th row of $\mathbf{P}$. Cross-multiplication of the first two equations with the third, yields the right-most form.

Equation pairs resulting from each given correspondence are stacked together into a homogeneous linear system. The solution is as usual obtained from SVD of the system matrix, with proper preconditioning, as the right-singular vector corresponding with the smallest singular value, and refined using Levenberg–Marquardt iterative optimization [Hartley and Zisserman 03, §7]. This is the basis of the often used calibration method of Tsai [Tsai 87], which among other things, also iterates the above sketched approach with lens distortion optimization.

In practice, the POSIT algorithm [DeMenthon and Davis 95] allows more efficient and robust pose estimation, when other camera parameters

[5] http://www.vision.caltech.edu/bouguetj/calib_doc/

are not needed. POSIT follows an iterative approach. It starts by estimating pose assuming an orthogonal projection model with scaling, rather than a full perspective model. The pose, obtained by making this assumption, is refined in a few, fast, iterations until good agreement is found with the actual observations. A linear algorithm for pose estimation was proposed in [Quan and Lan 99].

Example: Calibration of a Trifocal Camera Rig

Figure 1.3(top-left) depicts a camera rig consisting of a broadcast camera and lens with two auxiliary machine vision style cameras on its sides.[6] The auxiliary cameras are synchronized with the main camera, and provide additional views from which left-right stereo pairs can be generated in post-production, allowing control of convergence angle and baseline distance in post. These parameters need to be chosen differently, depending on the size of the screen the outcome is viewed on (ranging from mobile phone types of displays, over TV screens, to cinema). The lenses are wide-angle lenses, exhibiting distortion. The cameras were calibrated independently using the principles outlined in this chapter, as follows.

The camera rig is placed in front of a TV flat screen, on which a sequence of black-and-white patterns is displayed. The patterns consist of circular dots. The sequence of patterns reminds one of binary patterns used in structured light scanning, and allows to identify each dot in each camera. The dot centers form thousands of high-quality "labeled" rather than "anonymous" correspondences between the three captured views, also in cases in which patterns are not fully in view. Figure 1.3(bottom-left) shows the analysed pattern for the broadcast camera. Note how lens distortion is significant, even on the high-end broadcast camera lens used.

The dots form an equidistant grid in physical 3D space, and thus define a metric calibration world plane. Lens distortion is calibrated by first estimating the distortion center using a direct approach [Hartley and Kang 05]. Radial distortion parameters are obtained by iterative fitting of a homography to the image-to-calibration-grid correspondences minimizing deviation. Figure 1.3(bottom-right) shows that after lens calibration fitted and observed image coordinates correspond very well. Two observations of the calibration grid, from different angles (by moving the TV screen), were used.

The principal point was taken equal to the center of distortion, and pixel pitch and aspect ratio were taken from sensor specifications. Focal length was estimated using the IAC approach based on two observations of the metric plane defined by the calibration grid. In principle, a single

[6] http://www.20203dmedia.eu

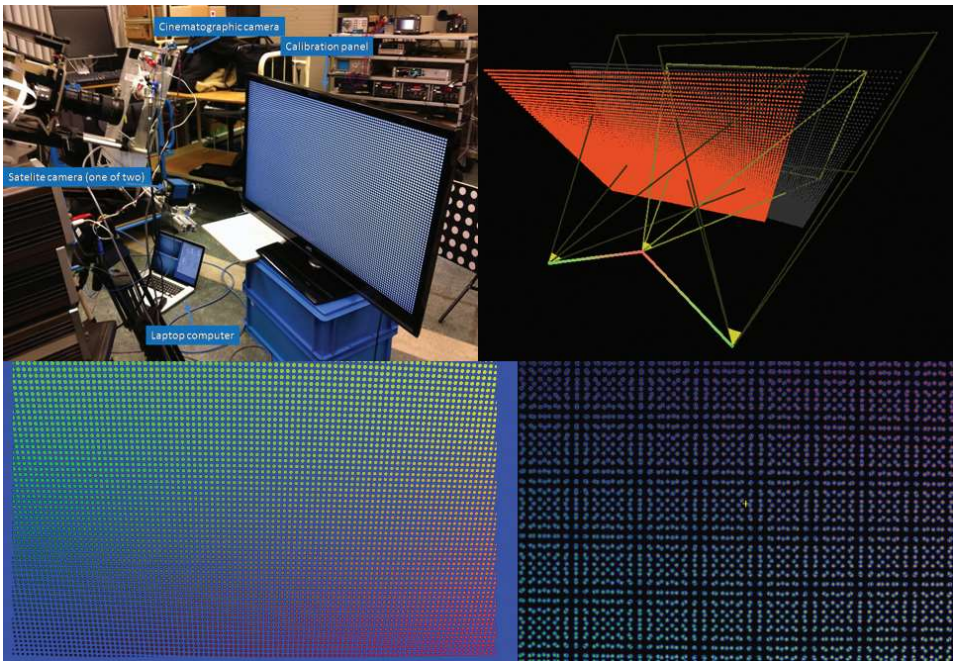


Figure 1.3: Trifocal camera rig calibration. Top-left: trifocal-camera set-up in front of computer-driven TV flat screen. Bottom-left: decoded calibration pattern on the central (broadcast) camera. Red corresponds with world-X, and green with world-Y coordinates. Note how lens distortion is significant, even on the high-quality broadcast lens used. Bottom-right: after lens distortion calibration, the fitted and observed image coordinates coincide very well. Top-left: calculated camera positions and orientations, and the relative position of the calibration grid, which was observed from two different angles.

observation would have done. The focal length of the auxiliary cameras was within 1% of lens specification.

Finally, the position and orientation of the cameras, and the calibration grid (viewed from two different angles), were calculated from the homographies and intrinsic matrices. The result, after refinement by iterative optimization, is shown in Figure 1.3 (top-right).

1.7 Summary

Going from uncalibrated, raw camera images to fully calibrated, meaningful, measured data is an essential prerequisite for many subsequent image

analysis tasks and visual computing applications. It must be noted, however, that the traditional image calibration pipeline described here, and in fact most computer vision algorithms, make implicit assumptions about the image acquisition process that hold only approximately for real-world photos and videos: For example, computer vision algorithms frequently assume that the *image exposure time* is indefinitely short. Physically, of course, any photo or video frame has been exposed for a finite period of time, potentially giving rise to motion blur [Sellent et al. 11]. Similarly, the *rolling shutter effect* of CMOS sensors and *streaking* of CCD sensors is frequently ignored. *Depth-of-field* caused by the, necessarily finite, lens aperture is also regularly not accounted for, nor is the fact that the widely used sRGB color signal constitutes a non-linear color model. Research into how to process images taken with commodity cameras to obtain physically meaningful measurement data has only just begun.

2
Stereo and Multi-View Video

Laurent Lucas, Céline Loscos, Philippe Bekaert, and Adrian Hilton

2.1 Introduction

The problem of multi-view acquisition relates to the capture of synchronized video data representing different viewpoints of a single scene. In contrast to video surveillance systems that aim to cover a large area with multiple cameras, the purpose here is to cover a single, often fairly restricted, physical space from multiple perpectives and to use the footage for 3D scene reconstruction to facilitate free-viewpoint rendering.

Depending on the final application, the number, layout, and settings of cameras can vary considerably. The most common configurations include lateral or directional camera arrays vs. global or omnidirectional multi-view cameras. In the first case, these devices provide multiple views, e.g., 2-views for binocular systems [Dubois 01, Peinsipp-Byma et al. 09] from close-together viewpoints, placed on the same side of a scene. They produce media adapted to stereoscopic displays. With regularly spaced cameras, autostereoscopic displays can be driven. In contrast, in omnidirectional systems multiple cameras are deployed around the target space. They are mainly designed for performance capture and free viewpoint applications. Finally, wide-baseline setups of multiple synchronized video cameras facilitate video-based, free-viewpoint rendering [Magnor 05].

Besides these purely video-based solutions, hybrid systems adding depth sensors to video footage are also currently being explored (Chapter 4). So far and regardless of the chosen capture device, all systems share the need to synchronize and calibrate (often even geometric and/or colorimetric corrections) information captured by different cameras, either online or as a post-process.

2.2 Multi-Video Capture Geometry

Configurations Overview

The generation of planar multi-stereoscopic views requires capturing with a synchronized multi-view camera [Perlin et al. 00, Dodgson 02]. Because such systems are intended to follow the natural binocular depth cue, it is not surprising that their design shows a striking similarity to the human eye. For example, the interaxial distance between the optical centers of the left- and right-eye camera lenses is usually chosen in relation to the interocular distance [Hill 53]. Furthermore, similar to the convergence capability of the eyes, it must be possible to adapt a camera to a desired convergence condition or zero parallax setting, i.e., to choose the 3D scene part that is going to be reproduced exactly on the display screen. This can be achieved by different camera configurations:

- "Toed-in" setup: with this setup the cameras are rotated inwards so that their optical axes converge at a single point. Since images have different trapezoid distortions, it is necessary to apply a systematic correction to enable the perception of 3D [Son et al. 07].
- "Parallel" setup: with this approach the cameras are parallel and the optical axes of the cameras converge at infinity. Without any post-processing shifting of the images to correct the zero parallax distance, objects at infinity will be cast at the surface of the display, and all other images will be cast in front of the display [Yamanoue 06]. Nothing appears behind the screen surface (no positive screen parallax).
- "Decentered parallel" setup: with this setup the cameras are again parallel. However, the images are shifted either in post-production, or in the camera by shifting two imaging sensors behind the lenses [Dodgson 02]. The shift of the images has the effect of changing the zero parallax distance (sometimes called the convergence distance).

Geometric principles described below are based on the latest model of camera. A more detailed description can be found in [Prévoteau et al. 10].

Viewing Geometry

The analysis of the characteristics of viewing geometry relies on a global reference frame defined in relation to the display device r centered in $\mathbf{o_v}$ (Figure 2.1(a)). The 3D display system mixes $n \times m$ images within its ROI (region of interest) with the dimensions L (width) and H (height). Each of these images (denoted by $\mathbf{I} = (i_1, i_2) \in \mathbb{N}_n \times \mathbb{N}_m$) is presumed to be

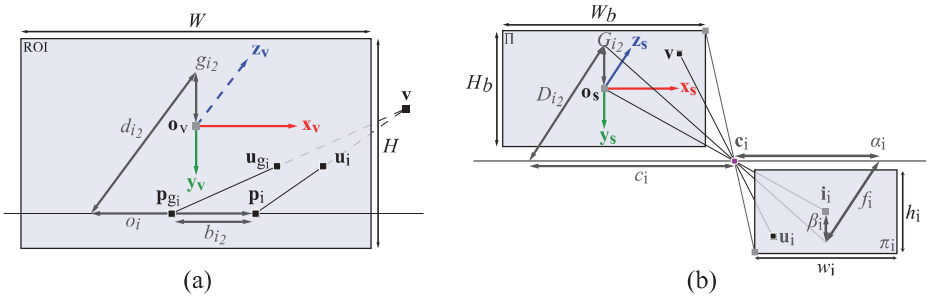


Figure 2.1: Characterization of geometry of: (a) the 3D multiscopic display using co-localized planar mixing; (b) the decentered parallel shooting setup.

correctly visible, at least from the preferred selected position $\mathbf{p}_i$. These positions are arranged on m lines parallel to the ROI lines situated at a distance of d_{i_2} from the system's region of interest.

Preferential positions are placed on these lines to ensure that the viewer, whose binocular gap is b_{i_2}, with the eyes parallel to the lines on the display, will have his right eye in $\mathbf{p}_i$ and his left eye in $\mathbf{p}_{g_i}$. The right eye in $\mathbf{p}_i$ will see image number $\mathbf{I}$ while the left eye $\mathbf{p}_{g_i}$ will see image number $\mathbf{I}_{g_i}$, knowing that $\mathbf{I}_{g_i} = \mathbf{I} - (q_{i2}, 0)$ where q_{i_2} represents the gap between image numbers composing coherent stereoscopic couples which are visible with a binocular gap of b_{i_2} at a distance of d_{i_2}. Combining the preferential positions of both the left and right eyes yields $\mathbf{p}_i = \mathbf{p}_{g_i} + b_{i_2}\mathbf{x}_v$ and $o_i = o_{g_i} + b_{i_2}$. The preferential position lines are placed on the vertical axis by g_{i_2} which represents the drop, i.e., the vertical gap between line i_2 of preferential positions and the center $\mathbf{o}_v$ of the ROI. When $m = 1$, the device does not create any vertical separation and any drop is acceptable a priori.

Supposing that the pixels $\mathbf{u}_i$ and $\mathbf{u}_{g_i}$ are stereoscopic homologues in the images $\mathbf{I}$ and $\mathbf{I}_{g_i}$, their perception by the right and left eye in $\mathbf{p}_i$ and $\mathbf{p}_{g_i}$, leads the viewer's brain to perceive the point $\mathbf{v}$ in 3D by stereopsis.

Shooting Geometry

The analysis of shooting geometry relies on the global shooting reference frame R, which is centered at the desired point of convergence $\mathbf{o}_s$ (which is also the center of the shared base Π in the scene) and is directed so that the first referential vectors are co-linear to the axes of the shared base Π in the scene and are therefore co-linear with the axes in the capture areas. In addition, the first axis is presumed to be parallel to the lines in the capture areas, and the second axis is parallel to the columns in these areas. The size of the shared base Π has the dimensions W_b and H_b. This reference

frame defines the position and direction of all the projection pyramids representing the capture areas by specifying the direction of observation $\mathbf{z}_s$ and the m alignment lines of the optical centers. In line with these principles, the $n \times m$ shooting pyramids are specified by:

- optical axes in the direction $\mathbf{z_s}$;
- the optical centers $\mathbf{c_i}$ aligned on one or several (m) straight lines parallel to the lines in the shared base and therefore the direction $\mathbf{x_s}$;
- rectangular capture areas $\pi_\mathbf{i}$.

The capture areas must be orthogonal to $\mathbf{z}_s$ and therefore parallel to each other and the shared base Π as well as the straight lines holding the optical centers $\mathbf{c}_i$ (which are defined by their distances to $\mathbf{o}_s$, D_{i_2} in relation to $\mathbf{z}_s$, G_{i_2} in relation to $\mathbf{y}_s$, and $c_\mathbf{i}$ in relation to $\mathbf{x}_s$). These capture areas are placed at distances of $f_\mathbf{i}$ in relation to $\mathbf{z}_s$, $\beta_\mathbf{i}$ in relation to $\mathbf{y}_s$, and $\alpha_\mathbf{i}$ in relation to $\mathbf{x}_s$ from their respective optical centers $\mathbf{c}_i$. Their dimensions are w_i and h_i. They are decentered in relation to their respective optical axes in the points $\mathbf{i}_i$ such that the straight lines $(\mathbf{i}_i\mathbf{c}_i)$ which define the target axes intersecting at the fixed point of convergence $\mathbf{o}_s$. The centers $\mathbf{c}_i$ and $\mathbf{c}_{g_i}$ must be on the same center line and spaced from B_i in relation to $\mathbf{x}_s$ ($\mathbf{c}_i = \mathbf{c}_{g_i} + B_i\mathbf{x}_s$ and $c_i = c_{g_i} + B_i$).

This kind of shooting configuration provides a depth perception on a multiscopic system with co-localized planar mixing with the possibility of a protruding as well as hollow image effect. However, this does not ensure that the perceived scene will not be distorted in relation to the original scene. The absence of distortion implies that the viewing pyramids are perfect homologues of the shooting pyramids (structures formed from Π with $\mathbf{c}_i$ as origins), i.e., they have exactly the same opening and deviation angles in both horizontal and vertical directions. Any deviation to this shooting and viewing pyramids homology involves a potentially complex distortion of the 3D image perceived in relation to the captured scene.

Distortion Analysis

According to the previous analyses of viewing and shooting geometries, it is possible to connect the coordinates $(\mathbf{x}_s, \mathbf{y}_s, \mathbf{z}_s)$, in the reference frame R, from the point $\mathbf{v}$ in the scene captured by the previously identified cameras with the coordinates $(x_1, y_\mathbf{i}, z_i)$ in the reference frame r of its homologue v_i perceived by an observer of the display device, placed in a preferential position (the right eye in $\mathbf{o}_i$).

The relation between the 3D coordinates of the scene point and those of its images perceived by a viewer may be characterized under homogeneous

coordinates by:

$$a_i \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} \mu_i & & \gamma_i & 0 \\ & \rho\mu_i & \delta_i & 0 \\ k_{i_2} & & 1 & 0 \\ \hline 0 & 0 & \frac{k_{i_2}(\epsilon_i - 1)}{d_{i_2}} & \epsilon_i \end{array}\right] * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The above equation can be seen as the analytic distortion model for observer position i which matches the stereoscopic transformation matrix given in [Jones et al. 01]. As such this model clearly exhibits the whole set of distortions to be expected in any multiscopic 3D experience, whatever the number of views implied or the very nature of these images (real or virtual). It shows, too, that these distortions are somehow independent from one another and may vary for each observer position i.

The above model exhibits some new parameters quantifying independent distortion effects. Those parameters may be analytically expressed from geometrical parameters of both shooting and rendering multiscopic devices. Their relations to geometrical parameters in and impact on distortion effects are as follows:

- $k_{i_2} = d_{i_2}/D_{i_2}$, control the global enlarging factor,
- $\epsilon_i = (b_{i_2}/B_i) \times (W_b/W)$, control the potential non-linear distortion which transforms a cube into a pyramid trunk according to the global scale factor $a_i = \epsilon_i + k_{i_2}\,(\epsilon_i - 1)\,\frac{Z}{d_{i_2}}$ possibly varying along Z,
- $\mu_i = b_{i_2}/\,(k_{i_2} B_i)$, control width over depth relative enlarging rate(s), or horizontal/depth anamorphose factor,
- $\rho = (W_b/H_b) \times (H/W)$, control height over width relative enlarging rate(s), or vertical/horizontal anamorphose factor,
- $\gamma_i = (c_i b_{i_2} - e_i B_i)\,/\,(d_{i_2} B_i)$, control the horizontal "shear" rate of the perceived depth effect,
- $\delta_i = (p_{i_2} B_i - G_{i_2} b_{i_2} \rho)\,/\,(d_{i_2} B_i)$, control(s) the vertical "shear" rate(s) of the perceived depth effect.

The previously detailed analysis of this model and its further inversion offers a multiscopic shooting layout design scheme acting from freely chosen distortion effects and for any specified multiscopic rendering device. Moreover, this model makes it possible to quantify those distortions for any couple of shooting and viewing settings by simple calculus based upon their geometric parameters.

2.3 Calibration and Synchronization of Cameras

In order to be able to relate imagery from multiple cameras with each other with the goal of 3D stereo viewing, view interpolation, 3D reconstruction, or other purposes, the cameras need to be synchronized and calibrated, both geometrically and photometrically.

A multi-camera system is geometrically calibrated as soon as all component cameras are calibrated. In principle, the methods explained in Section 1.6 suffice to calibrate also a multi-camera setup. However, correspondences between images of the different cameras, taken at the same time instance, can and shall be exploited for faster and better result. In this section, it is explained how image correspondences can be exploited for intrinsic and extrinsic calibration. Practical recommendations are given. The section concludes with suggestions concerning colorimetric calibration and synchronization.

Correspondences between images are typically provided automatically by feature point detectors and feature point description matching, such as SIFT. Feature detection, description, and matching are explained at an introductory level in text books such as [Szeliski 11], or, more in-depth in [Tuytelaers and Mikolajczyk 07].

Intrinsic Multi-Camera Auto-Calibration

In Section 1.6, the key to intrinsic camera calibration is to exploit certain scene features, such as orthogonality of 3D world lines and/or planes as observed in an image. These observations lead to constraints on a particular symmetric 3×3 matrix $\omega = \left(\mathbf{K}\mathbf{K}^\top\right)^{-1}$, called the *image of the absolute conic (IAC)*. The intrinsic camera parameters are obtained by decomposing ω using Cholesky factorization and Eq.(1.10).

The images of the absolute conic, by different cameras, at different positions and in different orientations, are related with each other. Exploiting these relationships allows to calculate the intrinsic (and extrinsic) parameters of the cameras in certain cases without the need to capture images of a calibration target, and even under completely uncontrolled circumstances.

It was explained in Section 1.6 that the vanishing points v_1 and v_2 in the image by a camera with intrinsic matrix $\mathbf{K}$, of world space points at infinity (direction vectors) D_1 and D_2, are related by $\mathrm{v}_2^\top \omega \mathrm{v}_1 = 0$, with $\omega = \mathbf{K}^{-\top}\mathbf{K}^{-1}$ the image of the absolute conic.

Now consider a second camera, with potentially different intrinsic matrix $\mathbf{K}'$, orientation $\mathbf{R}'$ and location C'. The vanishing points v'_1 and v'_2 in the image by this second camera, of the same world points at infinity D_1 and D_2, satisfies ${\mathrm{v}'_2}^\top \omega' \mathrm{v}'_1 = 0$. $\omega' = \mathbf{K}'^{-\top}\mathbf{K}'^{-1}$ is the IAC of the second camera.

The vanishing points by the two cameras are related by a homography $\mathbf{H}_\infty$, called the *infinity homography* of the second camera with respect to the first:

$$\mathrm{v}' = \mathbf{K}'\mathbf{R}'^\top \mathrm{d} = \mathbf{H}_\infty \mathrm{v} \quad \text{with} \quad \mathbf{H}_\infty = \mathbf{K}'\mathbf{R}'^\top \mathbf{R}\mathbf{K}^{-1}. \tag{2.1}$$

Note that camera location doesn't matter in imaging points at infinity.

The infinity homography transfers ${\mathrm{v}'_2}^\top \omega' \mathrm{v}'_1 = 0$ to a new relation between the vanishing points in the first view:

$${\mathrm{v}'_2}^\top \omega' \mathrm{v}'_1 = 0 \quad \Longrightarrow \quad \mathrm{v}_2^\top \mathbf{H}_\infty^\top \omega' \mathbf{H}_\infty \mathrm{v}_1 = 0$$

Since this relation holds simultaneously with $\mathrm{v}_2^\top \omega \mathrm{v}_1 = 0$, for any pair of vanishing points corresponding to perpendicular world space directions, $\mathbf{H}_\infty^\top \omega' \mathbf{H}_\infty$ must model the same conic as ω:

$$\omega \doteq \mathbf{H}_\infty^\top \omega' \mathbf{H}_\infty. \tag{2.2}$$

Each component of ω is linearly related to any component of ω' by above equation, up to a common scale factor. These relationships allow to transfer constraints on either one to the other.

The challenge, in practice, is to automatically determine the infinity homography $\mathbf{H}_\infty$ from image point feature correspondences in the presence of parallax, instead of from vanishing points. This is the basis of a multitude of auto-calibration methods, for rotating cameras, planar camera motion, turntable motion, moving stereo rigs, video sequences captured with a handheld video camera with zooming lens, and so on. [Hartley and Zisserman 03, §19].

Extrinsic Multi-Camera Calibration: Epipolar and Trifocal Constraints

Extrinsic calibration consists of locating camera optical centers and estimating the camera orientation. Methods for single-camera extrinsic (and full) camera calibration have been presented in Section 1.6. The key to extrinsic multi-camera calibration is to exploit epipolar and trifocal geometry.

Consider two cameras, seeing a world space point X at normalized image coordinates (camera space ray directions) $\hat{\mathrm{x}}$ respectively $\hat{\mathrm{x}}'$. Let the first camera define the world coordinate system (optical center at the origin 0, orientation matrix $\mathbf{I}$ is unity). Assume the second camera is at position C and in orientation $\mathbf{R}$, relative to the first. Non-coincident camera centers 0 and C form together with a non-collinear world space point X a plane,

named an *epipolar plane*. The cross product of $\mathrm{X} - \mathrm{C} \doteq \mathbf{R}\hat{\mathrm{x}}'$ and $\mathrm{X} - 0 \doteq \hat{\mathrm{x}}$ is perpendicular to the *epipolar axis* $\mathrm{C} - 0$, so

$$(\mathbf{R}\hat{\mathrm{x}}' \times \hat{\mathrm{x}}) \cdot \mathrm{C} = 0 \quad \Longleftrightarrow$$

$$\hat{\mathrm{x}}^\top \mathbf{E}^\top \hat{\mathrm{x}}' = 0 \quad \text{with} \quad \mathbf{E}^\top = [\mathrm{C}]_\times \mathbf{R} \quad \text{with} \quad [\mathrm{C}]_\times = \begin{bmatrix} 0 & -C_Z & C_Y \\ C_Z & 0 & -C_X \\ -C_Y & C_X & 0 \end{bmatrix} \tag{2.3}$$

The matrix $\mathbf{E}^\top$ is called the *essential matrix* of the first camera with respect to the second. The matrix $\mathbf{E}$, with $\hat{\mathrm{x}}'^\top \mathbf{E} \hat{\mathrm{x}} = 0$ for all corresponding $\hat{\mathrm{x}}$ and $\hat{\mathrm{x}}'$, is the essential matrix of the second with respect to the first. The essential matrix encodes the relative position and orientation of two cameras of any kind, with central projection model.

For rectilinear cameras, normalized image coordinates and image coordinates are related as $\hat{\mathrm{x}} = \mathbf{K}^{-1}\mathrm{x}$, and $\hat{\mathrm{x}}' = \mathbf{K}'^{-1}\mathrm{x}$', so that:

$$\mathrm{x}'^\top \mathbf{F} \mathrm{x} = 0 \quad \text{with} \quad \mathbf{F} = \mathbf{K}'^{-\top} \mathbf{E} \mathbf{K}^{-1}. \tag{2.4}$$

The matrix $\mathbf{F}$ is called the *fundamental matrix* of the second camera with respect to the first one. This relation states the well-known fact that the image x' of X in the second camera, must lie along a line $\mathrm{x}'^\top \mathrm{l}' = 0$: the intersection line of the epipolar plane of X with the second image. The lines' parameters are $\mathrm{l}' = \mathbf{F}\mathrm{x}$. Such lines are called *epipolar lines*. The same is true on the first image: $\mathrm{l}^\top \mathrm{x} = 0$, with $\mathrm{l} = \mathbf{F}^\top \mathrm{x}'$. All epipolar lines meet at a single point in an image, named an *epipole*. The epipole is the projection of the others camera position in the first camera's view. It shows the direction of travel of a moving camera.

The fundamental matrix encodes the complete projective geometry of two rectilinear cameras. Similar relations exist for non-rectilinear cameras, but epipolar lines become curves as dictated by the lens image formation model.

Given corresponding normalized image coordinates $\hat{\mathrm{x}}$ and $\hat{\mathrm{x}}'$, Eq.(2.3) is a homogeneous linear equation in the unknown coefficients of the essential matrix. The same is true for image coordinate correspondences and the fundamental matrix (Eq.(2.4)). Each correspondence yields such an equation. The matrix can be computed by stacking these equations together into a homogeneous linear system, which is solved using SVD with proper preconditioning and singularity constraint enforcement. False correspondences reported by automatic image feature detection and matching algorithms are filtered by means of a procedure named *random sampling consensus (RANSAC)*. The result is iteratively refined using the Levenberg–Marquardt non-linear optimization algorithm, and decomposed into relative camera position and orientation. These procedures are explained in detail in, for instance, [Hartley and Zisserman 03, §10 and 11].

It is important to realize that calibration, and 3D reconstruction, from epipolar geometry alone is fundamentally ambiguous by an arbitrary 3D homography. This can be understood as follows: if a camera matrix $\mathbf{A}$ images a space point X into x and $\mathbf{B}$ into x′, Then $\mathbf{AH}$ and $\mathbf{BH}$ image $\mathbf{H}^{-1}\mathrm{X}$ at exactly the same image locations x and x′. The fundamental matrices of any such pair of camera matrices are identical. Removing the projective ambiguity is possible only if cameras are also intrinsically calibrated, using for instance intrinsic multi-camera auto-calibration as outlined above.

Similar relations exist between three camera views, both in terms of image coordinates, and in terms of camera space ray directions (normalized image coordinates)

$$\sum_{i,j,k=1,2,3} \mathcal{T}_{ijk}\mathrm{x}_i\mathrm{x}'_j\mathrm{x}''_k = 0. \tag{2.5}$$

The $3\times3\times3$ table of numbers $\mathcal{T}$ is called a *tri-focal tensor*. There are 27 such relations for each triple-point correspondence, of which 4 are independent. Trifocal tensor computation is more involved than essential or fundamental matrix computation [Hartley and Zisserman 03, §16], but the resulting constraints are stronger: the locations of the image coordinates need to match exactly. With two-view constraints, image coordinates still have a 1D degree of freedom of moving along their epipolar line. For this reason, a stronger filtering of image correspondences results, as well as a more stable and accurate calibration and 3D reconstruction.

In practice, camera calibration can be obtained as a side result of 3D reconstruction software packages such as bundler[1] or VisualSFM.[2] These packages implement the principles outlined here, and refine calibration together with 3D position of image feature correspondences in one big non-linear optimization procedure called *bundle adjustment*.

Multi-Camera Matrix Factorization

A final noteworthy calibration principle is multi-camera matrix factorization. Consider a set of n world space points X_i, $i = 1 \ldots n$, which are all visible in a set of m cameras with full camera matrices $\mathbf{P}^j$, $j = 1 \ldots m$. The image of the i-th point by the j-th camera is $\mathrm{x}_i^j = \mathbf{P}^j\mathrm{X}_i$. Making the scale factors explicit, the matrix of image points can be written as:

$$\mathbf{W} = \begin{bmatrix} k_1^1\mathrm{x}_1^1 & k_2^1\mathrm{x}_2^1 & \cdots & k_n^1\mathrm{x}_n^1 \\ k_1^2\mathrm{x}_1^2 & k_2^2\mathrm{x}_2^2 & \cdots & k_n^2\mathrm{x}_n^2 \\ \vdots & \vdots & & \vdots \\ k_1^m\mathrm{x}_1^m & k_2^m\mathrm{x}_2^m & \cdots & k_n^m\mathrm{x}_n^m \end{bmatrix} = \begin{bmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \vdots \\ \mathbf{P}^m \end{bmatrix} \begin{bmatrix} \mathrm{X}_1 | \mathrm{X}_2 | \cdots | \mathrm{X}_n \end{bmatrix}.$$

[1] http://www.cs.cornell.edu/~snavely/bundler/
[2] http://ccwu.me/vsfm/

$\mathbf{W}$ is the product of a $3m \times 4$ matrix and a $4 \times n$ matrix, so it has rank 4. The $3m \times n$ matrix $\mathbf{W}$ can be decomposed in such product by means of SVD. The key to make this idea work in practice is in calculating the unknown scale factors on the fly from epipolar constraints [Sturm and Triggs 96].

Svoboda et al. used this as the basis for building a popular, complete and fully automatic multi-camera calibration MATLAB package [Svoboda et al. 05]. Metric calibration is obtained by exploiting multi-camera intrinsic constraints as explained in Section 2.3. Multi-image point correspondences may be generated by waving a laser pointer.

Example: Calibration of a Multi-Camera Studio for Human Motion Capture

Figure 2.2 illustrates the calibration of a multi-camera capture studio at the British Broadcasting Corporation BBC.[3] The studio contains twelve to fourteen broadcast TV cameras, capturing human motion from viewpoints all around. The cameras are genlocked, guaranteeing that they capture at exactly the same rate. The phase at which they sample is adjusted to be equal by observing a set of pulsed LEDs.

Broadcast cameras are photometrically calibrated by a white and black balance. For that they are pointed to a white sheet, and their "white balance" function activated. Next, their iris is closed completely, and the "black balance" function activated. This is a standard line-up procedure in TV broadcasting and provides pretty good results especially if cameras are of the same type. One potential problem specific to systems with cameras viewing a scene from a wide range of angles is that if the lighting is not uniform, and the white sheet used is not totally matte, then it might look slightly different when viewed from cameras in very different positions due to light reflections. In this case, the sheet may need to be moved or re-angled when white-balancing different cameras.

The cameras are calibrated geometrically using LED pole and multi-camera calibration tools developed by BBC,[4] based on bundle adjustment. Image correspondences are established by identifying images of a moving wand with pulsed LEDs mounted on a pole through the working space. The LEDs at the end have a different color, so observed LEDs are still suited for analysis even if the whole wand is not in view. This is particularly useful for more tightly zoomed-in cameras, like the bottom right example showing an actor's face in Figure 2.2. The top-right image in Figure 2.2 shows the LEDs observed by a camera. The bottom-right image in Figure 2.2 shows calibrated camera positions and orientations, as well as the position and orientation of the wand at analyzed instances.

[3] http://react-project.eu

[4] Developed by Julien Pansiot and Oliver Grau.

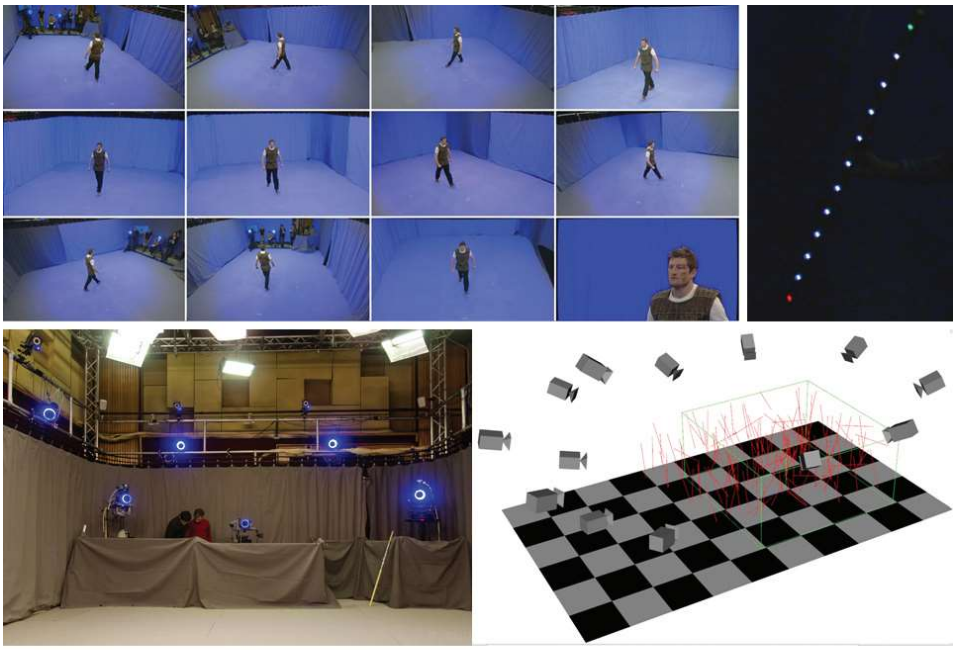


Figure 2.2: Multi-camera human capture studio calibration at BBC: top-left: views provided by the cameras of the setup; bottom-left: a view of the studio; top-right: camera view of the calibration LED wand; bottom-right: visualization of calibrated camera and wand positions and orientations.

Calibration in Practice

It pays off to spend effort in obtaining accurate calibration, to sub-pixel accuracy for 3D reconstruction and similar applications. Unfortunately, this is non-trivial even in controlled environments, and off-the-shelf software and approaches aiming for fully automatic calibration will often fail to provide the required accuracy if used as a "black box."

One problem is that several calibration parameters, such as position and orientation, or lens focal length and distortion parameters, are pseudo-dependent: small variations in different parameters cause almost identical effects and errors in one parameter but may be countered by errors in others. It may be tricky to decouple them when fitting everything together. Calibration procedures that proceed in steps, each addressing a set of independent parameters, can avoid this issue.

Many practitioners calibrate cameras using a mix-and-match approach: they combine the techniques (and software) presented in this text in ad-hoc

ways, often in part using their own implementations which are adapted to own-designed calibration grids and wands, camera setups, and so on. For instance, BBC have developed a wand with several pulsed LEDs on a line, as a more accurate alternative to laser pointer waving in Svobodas approach. In the example on page 20, a TV flat screen is used for producing sequences of calibration patterns producing "labeled" rather than "anonymous" calibration features.

It is beneficial to exploit all knowledge one can have about the setup. For instance, focal length may be known from fixed focal lens specifications; relative camera positions can be measured to a few centimeters' precision using a simple ruler in a fixed setup, or pan-tilt angles tracked with a MEMS sensor.

As usual in data modeling problems such as camera calibration, in addition to computed values, an indication of reliability also needs to be provided in the form of covariance estimates or confidence limits [Press et al. 07, §14].

Expressing camera calibration parameter values in human-understandable units helps to assess the plausibility of the result. If lens focal length, expressed in millimeters, is way off the lens indicator reading, for instance, or positions strongly disagree with what a ruler measures, there is most probably something seriously wrong with the calibration. The obtained results may explain world and image coordinates well for the particular calibration dataset, but there is no guarantee that the mapping will also be right for whatever other scenery.

Photometric Calibration

Many algorithms for processing multi-camera imagery assume color constancy, that is: an object seen by different cameras is observed with the same color. Obviously, this will only be the case for diffuse ("Lambertian") reflective objects, but even in the presence of non-diffuse objects, it is beneficial to make sure that colors from different cameras match each other as closely as possible.

This will be the case if the cameras are individually color calibrated and characterized as explained in Section 1.5. In practice, this means that a correct ICC profile is available and being used for each camera, and cameras are operating with equal settings, in particular with respect to white balancing.

If using color-calibrated cameras, or if cameras in a multi-camera setup are all of the same make and model, and operate with equal settings, good color consistency is often achieved using very simple means: a per-camera gain, compensating for slightly different lens apertures mainly.

In general, it pays off to color calibrate multi-camera setups using a Gretag–MacBeth color chart before actual shooting, even for calibrated cameras. The color chart allows to white balance the recording for the actual lighting being used, and calibrate consistently under that lighting.

For machine vision cameras, red and blue gain factors are set such that the neutral color patches of the chart appear as grayish as possible. Subsequently, the camera response on the neutral patches is inspected and a luminance curve calculated, linearizing response. Finally, a per-camera RGB color matrix is computed by non-linear optimization that equalizes the cameras response to the non-neutral color patches of the chart.

For small baseline setups, color matching by equalizing the cumulative histogram in the red, green, and blue channels is a fast and practical approach to ensuring color consistency between the cameras.

Even with the most sophisticated color equalization algorithms, however, there is no guarantee that all surfaces being captured will exhibit equal colors. Most materials in reality are not perfectly Lambertian, and colors will be different depending on illumination and view angle. Lens flares, due to interreflections of incident light inside camera lenses, is another major cause of color inconsistencies and local loss of contrast in practice. This is particularly noticeable with sunlight in outdoor situations, but also with studio spot lights.

Camera Synchronization

No matter how well cameras are calibrated for a static scene, geometrically and photometrically, without synchronization no sensible results will be obtained for moving scenes or moving cameras.

With the broadcast style of cameras, a genlock device will ensure that captured frames are transferred at the same pace for all cameras. This does not mean that cameras take pictures at the very same moment: there will in general still be capture phase differences between cameras. Broadcast cameras usually allow control of that phase. The phase can be calibrated by observing an old-style CRT monitor, or a set of fast flashing LEDs.

The machine vision type of cameras typically have an external trigger input, that allows control of the very moment that the camera shutter is opened. Driving it can be as simple as connecting these GPIO (general purpose IO) inputs to the lanes of a parallel port computer output, and sending byte codes to that parallel port. Or the GPIO exposure output signal of a master camera is hard wired to the external trigger input pins on the other cameras to slave cameras to the master, which is left running free at a desired frame rate.

This ensures that frames are captured simultaneously; however, different models of cameras will in general deliver frames with different delays.

Further delays are possible due to driver software and operating system issues on the capturing computer(s). In order to keep corresponding frames together, frames shall be tagged on the cameras by a time stamp or frame count in some way.

Broadcast cameras allow to embed SMPTE time code stamps with captured frames. SMPTE time code is received from a time code master clock.

Machine vision cameras allow to embed a camera clock and frame count. Camera clock drift will after a while cause different time stamps even for simultaneously captured frames. There may be one frame time difference as soon as after half an hour. Camera time stamps therefore must be synchronized by correlating with computer time or another external time source. Cameras need to be started appropriately to ensure consistent time stamp and frame count origin. Machine vision camera SDK documentation typically explains how to do this. A more powerful way to ensure consistent tagging of simultaneously captured frames from machine vision cameras is to feed additional IO signals and embed their state as frame metadata.

With consumer cameras, such as handycams, action cameras, and DSLR cameras, none of this is possible. Figuring out what frames are approximately captured at the same time is possible by inspecting the audio signal. Even with very cheap cameras, sound and image are usually recording synchronously to a sufficient extent. The audio signal will in general also allow to measure the time difference between different cameras shuttering for cameras of the same make and same firmware version. If audio and video are captured independently of each other, a clapper board and/or flash light or other sudden and fast movement or moving object will help to identify corresponding frames from different cameras. However, always keep in mind that, even when consumer cameras are started exactly at the same moment, using for instance their LANC input, there is no guarantee that their shutters will open and close simultaneously.

2.4 3D Cameras in Practice

A broad range of cameras is being used in 3D production. In case of binocular systems, capturing requires generally the use of two cameras connected by a rigid or articulated mechanical device known as a stereoscopic rig. Their optical design can be subdivided into two groups, multi-lens *vs.* single-lens systems (Figure 2.3). In both groups there is an even further distinction between single sensor and multiple sensor subclasses.

Single lens with single sensor setups follow Wheatstone's principle except that the camera is fixed to a slide bar. One image is taken at one point, the camera is moved along the bar to a fixed separation, then the second

Figure 2.3: Examples of 3D cameras and other devices: (a) STEREOTEC^{TM}side-by-side live rig; (b) 3DTV^{TM}Octocam camera array; (c) Panasonic^{TM}Lumix G 3D lens; (d) Panasonic^{TM}HDC-Z10000 stereoscopic twin-lens camcorder.

image is taken. Such a design allows for clear alignment and matching is much easier. However, they operate only for stationary scenes.

Conversely, single lens with multi-sensor setups involve problems of synchronization between sensors. Another drawback is cross talk between images since the resolution of the images is less than half of the sensor resolution. They also require adding components that need to be aligned and rely on minimal lens aberrations on the boundaries of the lens surface.

Multi-lens with single sensor setups combine views in a single sensor (Figure 2.3(c)). Merging all of the sensors makes it easier for matching and synchronization even if it does make it harder for the system to be versatile.

Finally, multi-lens with multi-sensor setups are probably the most common and most studied category of stereo cameras as they largely consist of two or more separate cameras in various orientations (Figure 2.3(a), (b) and (d)). Thus, cameras can be mounted at 90 degrees, looking through a half-mirror angled at 45 degrees or more simply, side by side.

2.5 Summary

Calibrating the setup of multiple cameras is a necessary prerequisite if image content is later to be inter-related across cameras and jointly processed. It can constitute a tedious and time-consuming procedure, especially when recording large scene volumes, e.g., outdoors. If small baselines suffice, e.g., for stereoscopic capture, rigidly mounted stereo or multi-view camera rigs can be used that are either already pre-calibrated or need to be calibrated only once.

3
Omni-Directional Video

Peter Eisert and Philippe Bekaert

3.1 Introduction

Omni-directional capture and display of scenes have a long tradition. Already in the 18th century, the English painter Robert Barker created large accessible panoramas showing a real scene from a particular viewing position, providing an immersive feeling of being there. The step from static scenes to omni-directional capturing of dynamic scenes was showcased at the World Fair of 1900 in Paris using 10 mechanical film cameras and projectors. During the last century, several commercial panoramic video systems followed, e.g., Disney's Circle Vision 360 or IMAX movie theaters. The advent of high quality digital cameras and projectors simplified the handling and processing of omni-directional video, i.e., the synchronization, calibration, warping, and blending, and made it widely applicable. Nowadays, there exists a large number of different high-resolution, multi-projection systems (Figure 3.1), that are used for immersive omni-directional cinema or presentation of panoramic multimedia events [Fehn et al. 07].

Besides the pure omni-directional reproduction of real-world scenes for entertainment purposes, many new applications have emerged. Today, omni-directional video displays are being used in simulators for professional training [Foote et al. 04], in modern digital planetariums for edutainment purposes [Lantz 07], as digital backlot for the cost-efficient representation of background scenes in movie productions [Schreer et al. 13] (Chapter 23), or they serve as an intuitive interface for large video collections [Tompkin et al. 13]. Many of these applications require synchronized high-resolution, omni-directional capturing of real-world scenes.

The simplest way of capturing omni-directional video is to use a single camera with a fisheye lens or a catadioptric system to image the scene hemispherically onto a single image sensor [Nayar 97, Krishnan and Nayar 08]. Although synchronization and stitching problems are avoided, acquisition resolution of such systems limits their use to mainly surveillance purposes. Instead, the focus here is on multi-camera systems that capture sub-parts of a scene in parallel. The multiple video streams are then stitched together again, resulting in a high-resolution panorama. The main

Figure 3.1: TimeLab: Panoramic 3D cinema with 7K resolution.

challenges in stitching and seamless blending of the individual parts consist of geometric and photometric corrections which are caused by non-exact focal point alignment of all cameras, differences in color response, view dependent scene appearance, and optical effects like lens flare. In addition, omni-directional scene capture typically requires high-dynamic range (HDR) technology (Section 1.5), because light sources as well as regions in shadow must be simultaneously imaged [Eden et al. 06].

In order to minimize geometrical corrections, the focal points of all cameras need to be as close together as possible. If the size of the cameras is small, a star-like configuration can be realized where all cameras face outward (Section 3.2). For larger cameras, or to avoid disparity correction, mirrors can be used that allow, at least theoretically, to place all focal points into the same position (Section 3.3). The creation of stereoscopic panoramic video, finally, faces the contradicting problem of providing large disparities for stereo viewing while requiring small disparities for stitching (Section 3.4).

3.2 Mirror-Free Panoramic Capture

In order to capture parallax-free omni-directional video with multiple cameras, ideally the cameras must be set up in such a way that their optical centers coincide. Unfortunately, this point typically lies somewhere within the lens objective, near the front of the lens for wide-angle lenses or more toward the back for lenses with long focal length. Consequently, it is

physically impossible to arrange several cameras such that their optical centers coincide.

The only practical way to ensure parallax-free omni-directional video capture is by using a single camera lens, or using multiple cameras with mirrors that are arranged in such a way as to effectively mirror the optical centers of all cameras into one virtual focal point (Section 3.3). On the downside, mirrors cause a multi-camera rig to be larger, heavier, and more delicate than a bunch of closely spaced cameras looking outward to capture a complete view of the world around. Moreover, it is not possible to realize a full 360×180 degrees omni-directional video capture rig this way without having the cameras or part of the mirrors being visible in the image itself.

Mirror-less camera rigs and tools: A common practice in high-end media and movie production today is to capture using a ring-shaped rig of digital still-image or movie cameras equipped with wide angle lenses, and to stitch the images together using commercial video stitching software like kolor[1] or open-source panorama tools like PTGUI.[2] Although by far the most panoramic and 360 video camera rigs being used are still custom-built,[3] some omni-directional cameras are also commercially available, e.g., the Point Grey Ladybug cameras,[4] or rig mounts for the popular GoPro action camera (Figure 3.2).[5]

Basic stitching: A convenient, real-time approach to stitch together multiple images into a single panorama is by virtually projecting the camera imagery onto the surface of a cylinder or, more accurately, a sphere. The sphere is viewed from its center. The virtual projection loci correspond to the relative positions of the cameras in the rig. Orientation, field of view, lens distortion of the virtual projectors must all match that of the real cameras. The virtual projection can be implemented conveniently using projective texture mapping, e.g., using OpenGL. The same shaders allow to perform feathering with alpha values based on the distance to the edge of the camera images raised to a power allowing to control the sharpness of the blending. This produces perfect stitching at a single distance to the spectator only. Everything closer or further away will appear doubled to some extent. However, the distance can be controlled at will. This basic stitching approach already yields useful results in many practical cases, given judicious camera placement and parameter control (sharpness and distance).

Calibration: Before stitching can be performed, the relative orientation and position of the cameras in the rig as well as lens properties

[1] www.kolor.com
[2] www.ptgui.com
[3] http://www.radiantimages.com/blog/9-red-hot-chilli-peppers
[4] www.ptgrey.com
[5] www.360heros.com, www.freedom360.us

Figure 3.2: Mirror-less panoramic and omni-directional multi-camera rigs: 360-hero rig with GoPro cameras (top left), Point Grey Ladybug (top second from left), and other custom-built rigs (Hasselt University, Belgium).

(i.e., focal length, principal point, lens distortion; cf. Section 1.6) must be precisely calibrated. In principle, generic camera calibration methods and procedures can be used (Section 2.3). For improved results, additional a priori knowledge can be exploited. For example, the relative orientation and position of all cameras is known from rig design. Fine-mechanical constructions like camera rigs are typically realized to a precision of one tenth of a millimeter in position and one tenth of a degree in orientation. Also, typically fixed focal distance (prime) lenses are used in omni-directional and panoramic camera rigs whose focal length match lens specifications to one tenth of a millimeter. If the result of the calibration procedure disagrees by more than the afore-mentioned tolerances from the mechanical rig design or lens specifications, most probably the calibration procedure is overfitting or suffers from pseudo-dependencies between parameters, such as orientation and principal point or focal length and lens distortion. The most important unknown parameters in calibration are the location of the optical center within the camera-lens system, the principal point (image coordinate at which the optical axis of the lens intersects the camera sensor),

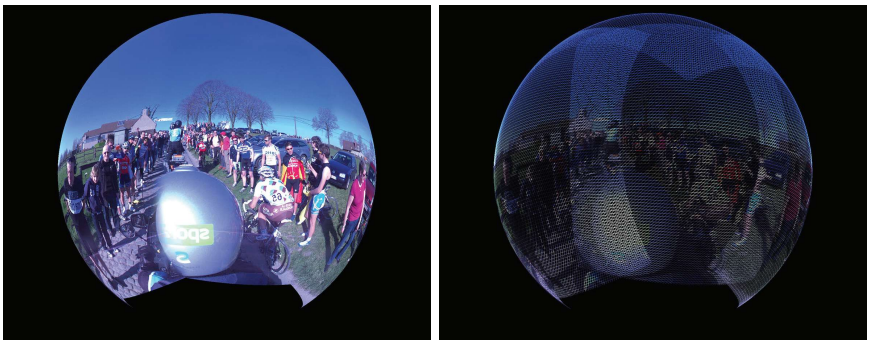


Figure 3.3: Basic stitching: using projective texturing, camera images are mapped onto the surface of a sphere. The virtual spectator is located at the center. The right image shows the projections as wireframes to visualize the composite of the stitched result at the left. This simple method produces exact results at one fixed scene distance only. Everything closer or farther away shows ghosting artifacts, e.g., the motorcyclist helmet.

and lens distortion parameters. Lens distortion can be accurately measured by observing straight lines or rectangular grid patterns (Section 1.6). The lenses' principal points can usually be determined accurately by shooting a distant scenery in which the distance between camera centers no longer causes noticeable parallax. Once geometric relationships are established, photometric properties like color reproduction and lens vignetting must be measured for each camera and adjusted (Section 2.3).

Advanced stitching: A good starting reference for studying warping and stitching for panoramic and omni-directional photography and video is [Szeliski 06]. It describes the basic theory as well as a full range of methods on how to align and register component imagery and to compose them properly. Ideally, a stitching algorithm transforms the set of captured images in such a way that the resulting composite is identical to a panorama image captured with an imagery camera rig with exactly coincident optical centers. To correct for local parallax in small-baseline camera rigs, one promising approach is to build upon stereo- and multiple view interpolation [Scharstein and Szeliski 02, Rogmans et al. 09]. By interpolating overlapping camera images novel views from coincident locations can be synthesized, resolving any parallax issues. In theory, the depth estimates resulting from such algorithms may also be used to generate stereoscopic views and allow for 3D reconstruction, just as with conventional stereo camera rigs (Section 17.3). In order to provide high effective image resolution, however, practical mirror-less panoramic and omni-directional multi-camera rigs feature only little overlap between camera views. Still,

Figure 3.4: As long as camera placement and algorithm parameters (sharpness and distance) are carefully selected, basic stitching already provides acceptable results in many practical cases, even in the presence of close-by scenery and large depth variations.

view interpolation helps to reduce parallax artifacts in the overlapping area [Szeliski 96, Kang et al. 04, Qi and Cooperstock 08, Adam et al. 09]. In practice, a remaining obstacle is the rather unpredictable and ungraceful failure characteristics of current view interpolation algorithms. There is steady progress in this area, however (Section 17.2), and it may become the preferred approach to omni-directional video capture in the long run.

Yet another approach consists of carving images along irregular seams [Ozawa et al. 12]. While this works well for still panoramas, optimal seam carving often results in objectionable flickering when applied to video.

3.3 Mirror-Based Panoramic Capture

For star-like camera arrangements as described in the previous section, the focal points of all cameras lie on a circle whose minimal radius is determined by the physical size of the cameras and lenses used. This inevitably leads to parallax effects that make stitching and blending of the overlapping areas difficult, especially for objects close to the cameras. In principle, these problems can be avoided by using a pyramidal mirror system [Tan et al. 04, Hua et al. 07, Weissig et al. 12] (Figure 3.5). By correctly placing

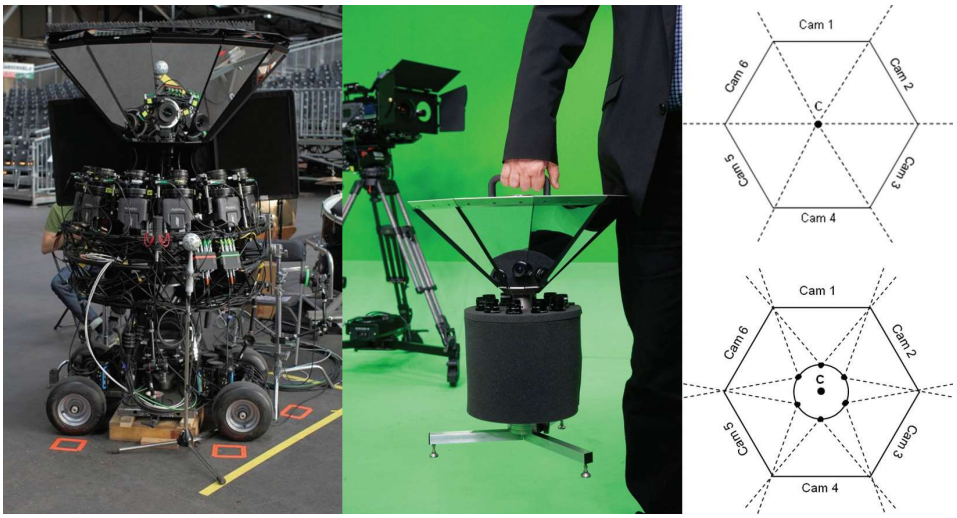


Figure 3.5: Prototype of a mirror-based omni-directional cameras for 3D panorama acquisition [Weissig et al. 12].

the cameras under a rig of differently oriented planar mirrors, the optical centers of all cameras can be made to virtually coincide at the same 3D position, independent of camera size. This allows for the use of high-end cameras, e.g., six ARRI Alexa digital cinema cameras in the OmniCam prototype (Figure 3.5 left). The system can record video panoramas at a resolution of 6K x 2K pixels and at 16 bits of dynamic range. A less bulky and mobile version consists of ten micro HD cameras capturing 360° panoramas and a vertical viewing range of 60° at 10K x 2K pixels (Figure 3.5, middle). For applications that require larger viewing angles, two mirror pyramids can be placed on top of each other [Tan et al. 04, Hua et al. 07]. By tilting the mirrors less than 45°, the focal points of upper and lower cameras can still be brought into alignment, at the price of not fully exploiting full sensor area anymore due to trapezoidal distortion.

While the mirrors allow for perfect alignment of all cameras' optical centers, the resulting individual images do not feature any overlapping area anymore which makes seamless stitching and blending of the individual tiles almost impossible. Sufficient overlap of adjacent images can be created by shifting the focal points of the cameras slightly outward (Figure 3.5 right). This off-center placement enables the adjustment of image overlap at the expense of introducing small image disparities that, however, can be corrected for as described in Section 3.2. The OmniCam prototype system (Figure 3.5), for example, employs a radial shift of 5 mm that results in an image overlap of about 10 pixels. It allows for parallax-free stitching of scenes with a depth range from about two meters to infinity.

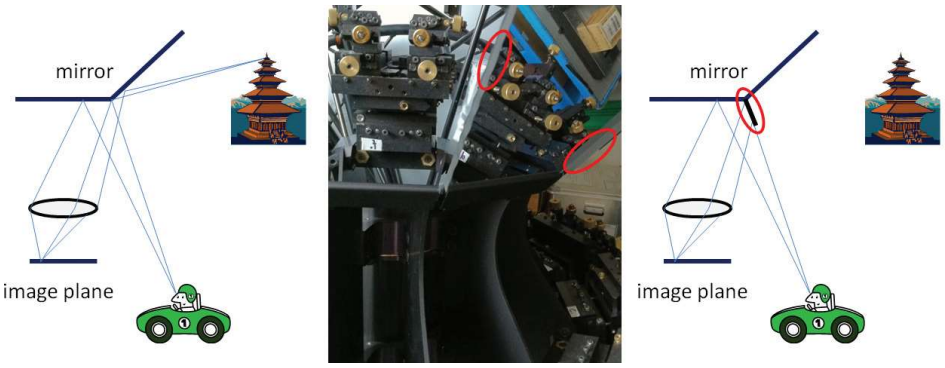


Figure 3.6: Crosstalk at mirror borders can lead to ghosting artifacts. Small blinds between mirror segments help prevent crosstalk.

One practical problem of the pyramidal mirror setup is the introduction of small blurring artifacts at the image borders between mirror segments that become visible at high resolution. Along the mirror edges, due to the finite aperture of the camera lenses a camera captures reflections from both mirror segments (Figure 3.6). By integrating light over two mirror planes, object points from different directions may be projected onto the same pixel resulting in blurring or ghosting. This effect can be reduced by closing the aperture such that the mirror is in focus, but this would significantly reduce the light falling onto the sensor and make capturing in dark environments difficult. Instead, small black blinds can be added between the mirror segments to avoid crosstalk (Figure 3.6). The blinds mask the neighboring mirror segments from the view of each camera such that each camera integrates only over one mirror tile and the black blind, leading merely to some darkening of edge pixels. Since this darkening occurs in the overlapping area where the final image is additively blended from two segments, it remains unnoticeable if photometric calibration and correction is properly performed. With all processing steps diligently applied, the resulting panorama is indistinguishable from that obtained with a single-chip panorama camera (Figure 3.7).

3.4 Stereoscopic Panoramic Capture

In contrast to the 2D case, capturing stereoscopic panoramic video is much more challenging due to the conflicting requirements of little or no disparity for seamless stitching and large disparities for stereoscopic viewing experience (Chapter 18). A star-like camera arrangement does not work well for stereoscopic panorama capture (Figure 3.8 left). The unavoidable baseline

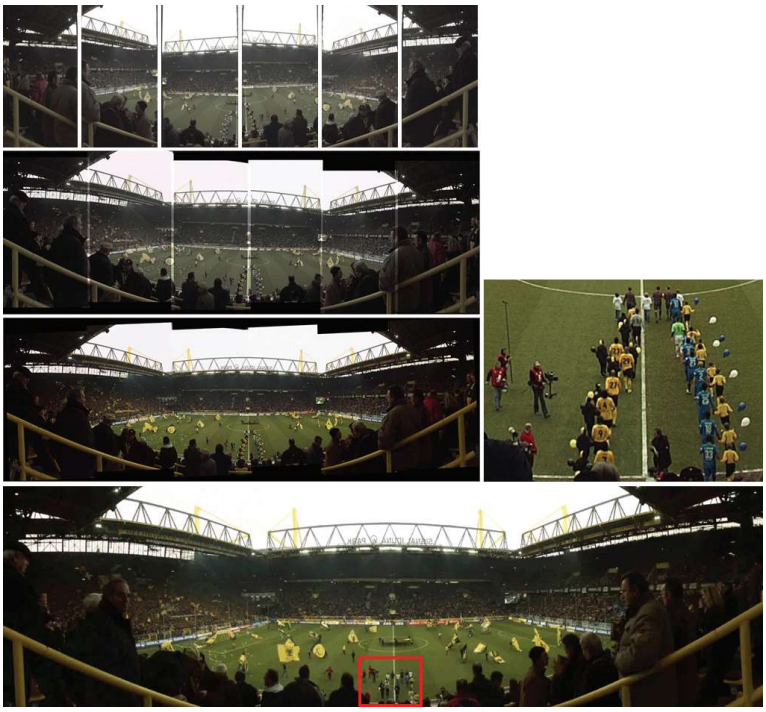


Figure 3.7: Subsequent processing steps necessary to create a panorama image using a multi-camera capture rig. From top to bottom: original camera views; geometric correction; photometric correction and blending; final cut-out of panorama with additional close-up views illustrating the high resolution [Weissig et al. 12].

B between adjoining cameras for the left and right view is smaller than the baseline S between neighboring camera pairs. This renders stitching very difficult and is likely to introduce annoyingly visible disparity artifacts between camera pairs. The problem becomes less severe when using a larger number of smaller cameras, but the unfavorable discrepancy between baselines B and S persists.

One common method to overcome this problem is to eccentrically rotate two slit cameras (or one regular area camera) and to concatenate image columns over time to stitch a panorama. Since the virtual camera positions are extremely close together, no disparity problems occur during stitching. This is, for example, exploited in omni-stereo or concentric mosaic capturing [Peleg et al. 99, Shum and He 99, Peleg et al. 01] and has been successfully employed in the creation of high-resolution stereo panoramas [Richardt et al. 13]. The disadvantage of the rotating camera setup is its inherent limitation to static scenes only. This problem has been

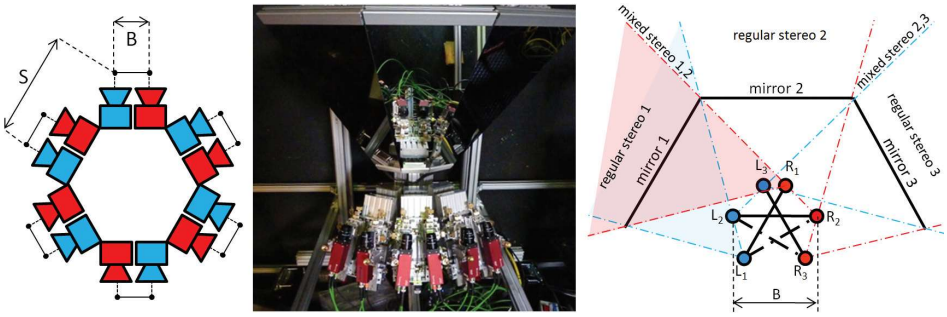


Figure 3.8: Left: Illustration of the conflict between desired stereo baseline B and unwanted parallax S between neighboring camera views for the mirror-free design. Middle: 3D omni-directional camera with 3 camera pairs for 180° capture. Right: Geometric design of a stereoscopic omni-directional camera.

addressed by [Belbachir et al. 12] who developed a high-speed rotating camera that can capture stereo panoramic video at 10 fps and 1800 pixels for 360° cylindrical panoramas. While the approach can, in principle, be extended to even higher frame rates and resolution, the resulting very short shutter times require extremely bright scenes.

Instead of sequentially scanning the scene, Peleg et al. used a sophisticated system of spiral lenses and parabolic beam splitters in order to simultaneously capture two panoramas [Peleg et al. 00]. A system relying on a less complex optical system is proposed in [Schreer et al. 12]. The 3D OmniCam shown in Figure 3.8 is based on a pyramidal mirror system with 3 camera pairs capturing 180° (or 6 pairs for full 360°). Similar to the 2D case, the mirrors position the focal points of all cameras along a small circle. The advantage of this approach compared to the star-like setup is that disparity for stereoscopic viewing comes from cameras on the circle opposite to each other while stitching is performed between the much closer neighboring (virtual) camera views (Figure 3.8 right). $L1$ and $R1$ are the virtual focal points for a pair of cameras looking through mirror segment 1. This design makes sure that distance $\overline{L_1R_1}$ is larger than $\overline{L_1L_2}$. The real cameras of baseline $B = 6$ cm are toed-in such that their optical axes cross on the mirror surface. This ensures maximal overlap of left and right views on a mirror segment. Still, in the sector indicated "mixed stereo 1 2" only one camera (L_1) captures the scene via mirror segment 1. For this region, the corresponding image data for the right view must be taken from R_2 looking through segment 2. As illustrated in Figure 3.9 (left), the final left and right panoramas are stitched together such that there are regions that originate from left and right cameras of the same mirror segment, or, for

Figure 3.9: Left: Illustration of the different regular and mixed stereo segments. Right: Simulation of left and right panorama stitching using a geometrical setup [Schreer et al. 12].

the mixed stereo case from neighboring mirror segments. Similar to the 2D setup (Section 3.3), the cameras' focal points are shifted slightly outward such that the baselines do not intersect the mirror center. This again leads to an overlap in viewing angles such that seamless stitching across mirror segments can be accomplished. The shift must be adjusted such that the regular baseline $\overline{L_1 R_1}$ is the same as the baseline $\overline{L_1 R_2}$ for a mixed stereo pair. With this mirror setup, the conflicting requirements on disparity for stitching and viewing can be elegantly solved, enabling the capture of high-quality stereoscopic omni-directional video [Schreer et al. 12].

3.5 Summary

The advent of small digital cameras has enabled the development of cost-efficient high-resolution, omni-directional video capture devices. High-quality video panoramas can reproduce arbitrary views of a scene from a fixed vantage point. Omni-directional video has many interesting applications like background representations in movie productions or in format-agnostic productions for different displays and devices. 2D omni-directional video is relatively straightforward and can be realized compactly by a star-like camera arrangement. In this setup, due to the physical size of the cameras parallax effects cannot be avoided and have to be corrected for by appropriate stitching algorithms. For very high quality results, bigger cameras and lenses are needed, increasing the disparity problem. A mirror rig allows for virtually placing all focal points at the same position, enabling, in principle, parallax-free stitching. In practical scenarios, however, focal points must be radially shifted somewhat to create overlap-

ping image regions between mirror segments. Stereoscopic omni-directional video is much more difficult to acquire due to the conflicting requirements of little disparity for seamless stitching while needing large disparities to enable stereoscopic viewing experience (Chapter 18). For static scenes, a rotating camera can sequentially scan the environment. For stereoscopic panoramic video capture, more sophisticated capture setups are necessary. One possibility is again to use a mirror rig to place all focal points on a ring around the center, leading to large baselines of opposite camera pairs for stereo impression and small baselines between neighboring cameras for stitching. This enables capturing convincing high-resolution, stereoscopic, omni-directional video while avoiding parallax artifacts.

4
Range Imaging

Andreas Kolb and Fabrizio Pece

4.1 Introduction

A vast number of applications benefit from or even require geometric information acquired from real environments, such as cultural heritage, virtual and augmented environments, human machine interaction, safety in an industrial or automotive context, to name just a few.

The notion of range imaging subsumes contact-free techniques for acquiring per-pixel distance information with respect to a current pose using sensors comprising a 2D array-like arrangement of sensor elements (*pixels*). Even though a single acquired frame formally results in a 2.5D information, a large number of applications use a series of these 2.5D datasets in order to recover a complete 3D model; therefore the data delivered by range sensing systems is usually denoted as 3D data.

From the technical perspective, there are two basic principles for range imaging, namely time-of-flight (ToF) and triangulation-based methods. The ToF technology is based on measuring the time that light emitted by an illumination unit requires to travel to an object and back to the sensor array. This is used in LIDAR (light detection and ranging) scanners for high-precision, point-by-point distance measurements. In the last decade, this principle has found realization in microelectronic devices, i.e. chips, resulting in new range-sensing devices, the so-called *ToF cameras*. Triangulation-based methods, on the other hand, utilize the disparity effect, i.e., an object is displaced in the image plane as the observing 2D-camera is moving in (approximately) lateral direction. As the amount of displacement relates to object distance as well as to camera displacement (*baseline*), the distance can be reconstructed in case the baseline is known. Triangulation-based systems can be realized as *passive devices* resulting in the *stereo vision* or *multi-view vision* approaches of Chapter 8. Light-field cameras can also be used for passive triangulation (Chapter 5). *Active triangulation-based systems*, on the other hand, realize the triangulation principle using (at least) one active illumination device instead of passive 2D cameras. The illumination unit, e.g., a projector or a laser sheet, is used to "code" the spatial directions from the position of the active device in

such a way that the observing 2D camera can identify these directions in the image plane. Again, the two directions from the active and the passive device together with the baseline form a triangle in space from which the distance to the observed object point can be deduced. Specific instantiations of active triangulation methods are structured light scanners which use a beamer to code the spatial directions by one or several patterns, and laser triangulation which uses a sweeping laser sheet to allow for a row-by-row acquisition of the object's geometry.

The recent trend in range imaging is clearly toward very fast and low-cost off-the-shelf 3D imaging acquisition systems which are able to acquire range images in video frame rate, i.e., with 30 FPS or more. The two currently most prominent representatives are discussed in this chapter: *ToF-cameras* and the Kinect^{TM}sensor, issued by Microsoft Corp. in conjunction with the XBox 360 game console. Recently Microsoft issued a new Kinect^{TM}version which is based on ToF imaging technology. At the time of writing Microsoft is preparing to issue a new Kinect 2^{TM}version which will use ToF imaging technology.

4.2 Structured Light Cameras—Kinect^{TM}

Even though the principle of structured light (SL) range sensing is comparatively old, the launch of the Microsoft Kinect^{TM}in 2010 as interaction device for the XBox 360 clearly demonstrates the maturity of the underlying principle.

Technical Foundations SL cameras, such as the first version of the Microsoft Kinect^{TM}, typically employ an IR laser projector combined with a monochrome CMOS camera which captures depth variations by analyzing the way a projected pattern deforms when striking an object surface in the scene. This is the same principle of structured light 3D scanners. Historically, structured light scanners operated at low frame-rate. Recently though, Liu et al. have made a major speed breakthrough in 3D laser scanning introducing a technique that can reach data processing performance of 120 Hz [Liu et al. 10a]. By utilizing the binary defocusing technique, Zhang and Huangeven have shown that even higher working frame-rate, well over 500 Hz, can be achieved without loss of precision [Zhang et al. 10b].

While the general principle to realize SL systems is based on analyzing the deformation of a projected IR light pattern, different approaches to achieve this have been investigated in the literature. Zhang and Huang propose a real-time SL system which runs on specialized hardware [Zhang and Huang 04]. The authors present a real-time scanner that uses digital

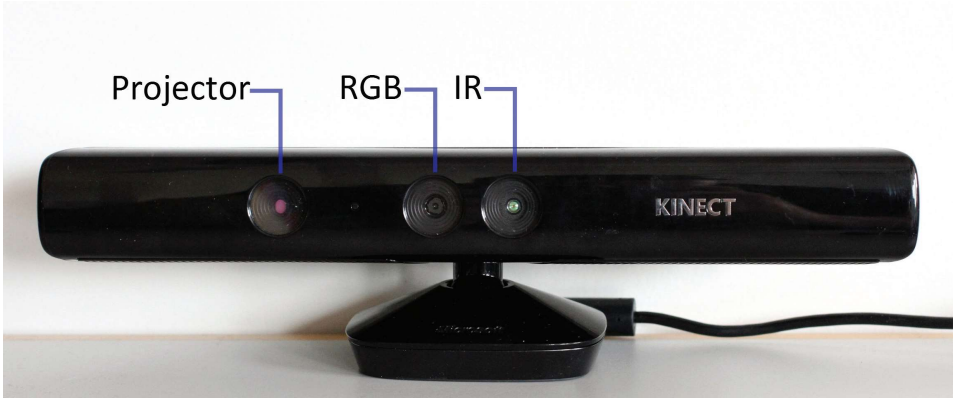


Figure 4.1: Sensor placement within a Kinect™sensor. The baseline is of approximately 7.5cm.

fringe projection and phase-shifting techniques to capture, reconstruct, and render high-density details of dynamically deformable objects such as facial expressions. The proposed system is capable of working at high rates, up to 40 Hz. Scharstein and Szeliski are able to capture high-resolution depth maps of complex scenes by combining multiple SL projectors [Scharstein and Szeliski 03]. [Zhang et al. 02] propose a variation of classic structured light algorithms that uses a pattern of stripes of alternating colors to match observed edges in the scene and to recreate 3D shapes; similarly, [Fechteler et al. 07] propose a fast and high-resolution 3D scanner that recreates textured 3D shapes from just two images. Hall-Holt and Rusinkiewicz introduce a SL method to obtain real-time structured light range scanning based on a new set of illumination patterns [Hall-Holt and Rusinkiewicz 01]. Such patterns are based on coding the boundaries between projected stripes. The stripe boundary codes allow range scanning of moving objects at 60 Hz with 100 μm accuracy over a 10 cm working volume. The system uses a standard video camera and digital light processing projector (DLP) to produce dense range images.

Kinect™ technology is based on the classic SL approach. The unit comprises two cameras, one RGB and one IR, and one laser-based IR projector (Figure 4.1). The IR camera and the IR projector form a stereo pair with a baseline of approximately 7.5 cm. The IR projector sends out a fixed pattern of light and dark speckles. The pattern is generated from a set of diffraction gratings that are designed to lessen the effect of the zero-order propagation, i.e., to avoid a centered bright dot [Zalevsky et al. 07].

Depth calculation is performed by triangulating the known pattern emitted by the projector, that is stored on the unit. For each new frame,

depth is estimated at each pixel p_i by sliding a correlation window on the recorded IR frame. The window is typically small (9×7 or 9×9 pixels). It is used to compare the recorded pattern at p_i with the corresponding stored pattern. The best match gives an offset from the known depth, in terms of pixels, also known as disparity. The device performs an interpolation of the best match to get sub-pixel accuracy to $\frac{1}{8}$ of a pixel. Given the known depth of the stored pattern and the disparity value, an estimated depth for each pixel is calculated by triangulation.

Challenges Working with the Kinect$^{\text{TM}}$sensor is relative easily as plenty of drivers and resources are provided for a variety of operating systems. However, challenges arise when trying to calibrate the internal sensors to obtain depth-plus-color information (i.e., RGBD pixels), when the systematic depth error needs to be calibrated, or when multiple units have to operate together.

Camera Calibration The IR and RGB cameras are separated by a small baseline of approximately 2.5 cm horizontally. Hence, if color and depth information are needed for each pixel, calibration of one sensor to the other has to be performed. As the IR camera can be considered a variation of a gray-scale camera, the classic checkerboard method (Section 1.6) can be employed for extrinsic calibration. Such calibration routines assume that the intrinsic parameters of the two sensors are known; if this is not the case, intrinsic calibration can be performed by using a zero-distortion model for the IR camera, and a distortion and de-centering model for the RGB camera. Typical values for the translations between the two cameras will be close to zero along Y and Z axis, and approximately 2.5 cm on the x axis. The rotation between the sensors is minimal, with values smaller than 1 degree.

The above solution, while based on an established calibration method for which a variety of tools are readily available [Bouguet 08, Bradski 00], is unfortunately prone to error as it relies on accurate matches of image features across both cameras. Accurate matches between RGB and IR cameras, however, is surprisingly difficult to establish. Therefore, other calibration routines have been designed for the Kinect$^{\text{TM}}$. [Herrera C. et al. 12] present an algorithm that simultaneously calibrates two RGB cameras and a depth camera as well as the relative pose between them. The method is designed to calibrate a Kinect$^{\text{TM}}$ unit to an external RGB image, and it proves to be more accurate and reliable than the classic checkerboard method. The algorithm requires only a planar surface to be imaged from various poses. The calibration does not use depth discontinuities in the depth image which makes it flexible and robust to noise. In addition to camera calibration, the authors presented a new depth distortion model for the

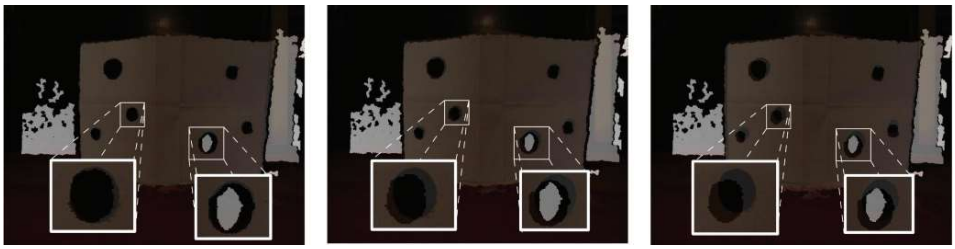


Figure 4.2: Comparison of two calibration solutions' accuracy. An object with holes is acquired by a Kinect™sensor, and the RGB image is overlaid on the depth-map after calibrating the two cameras. The [Raposo et al. 13] method yields accurate results with small misalignment (left). The [Herrera C. et al. 12] solution introduces significant misalignment without distortion correction (center). The misalignment becomes even more severe if distortion correction is applied (right).

depth sensor which can be used to calibrate the unit systematic error. [Raposo et al. 13] modify the above work to employ only 6–10 disparity-image pairs of a planar checkerboard pattern, dramatically speeding up the calibration time and requiring only a few input images. Figure 4.2 shows the comparison of the two methods.

A different approach is introduced by [Zhang and Zhang 11]. The authors present a maximum likelihood solution for the joint depth and color calibration based on two principles. First, co-planarity of the checkerboard points in the depth image is enforced. Second, additional point correspondences between the depth and color images are manually specified or automatically established to help improve calibration accuracy. Finally, a variation on intrinsic calibration is presented by [Teichman et al. 13]. The author presents a new generic approach to the calibration of depth sensor intrinsics based on simultaneous localization and mapping (SLAM) [Smith and Cheeseman 86]. In particular, no specialized hardware, calibration target, or hand measurement is required, making the calibration routine completely unsupervised. Compared to the supervised calibration approach, the proposed technique requires only a few minutes of data input from unmodified, natural environments. This is particularly advantageous in situations where humans cannot be involved, e.g., to hold checkerboards in front of the camera, such as for a robot that needs to calibrate automatically while in the field.

Systematic Error Correction Similar to most range cameras, the Kinect™ suffers from systematic error in depth estimation. Khoshelham and Elberink analyzed the accuracy and resolution of the sensor depth data, showing that the systematic error is generally smaller than 3 cm, but that

it increases on the periphery of the sensor or when depth measurements are collected further away from the unit [Khoshelham and Elberink 12]. Interestingly, the error seems also to be stronger when depth measurements are collected close to the camera sensor [Smisek et al. 11]. There are several approaches to handle the systematic error, including the one presented in [Smisek et al. 11]. [Herrera C. et al. 12] proposed a distortion model to correct the systematic unit error. A different approach is introduced by [Yamazoe et al. 12]. The general principle beyond their calibration routine is that, as the SL principle is based on both emitter and receiver, the intrinsic parameters of both the IR camera and projector should be taken into account. Hence the authors present a depth correction model that is based on joint estimation of depth-camera and projector intrinsic parameters, achieved by showing only a planar board to the depth sensor.

Multiple Units Integration Combining several Kinect^{TM}cameras is non-trivial due to potential interference problems given by multiple IR patterns projected into the scene. To combat this, one can carefully align multiple units to avoid IR overlaps, but this requires a tedious manual calibration. A more general solution, based on constant shake of the units, has been recently presented by [Butler et al. 12]. The authors propose to associate to each unit a motor with an offset weight. The motor shakes the camera, and consequently also the IR projector and camera are moved. As the shaking is constant for both sensors, the depth estimation algorithm still works reliably for the single unit. However, from the viewpoint of another Kinect^{TM}, the pattern of the other projector moves around and interferes with its own pattern only for a small amount of time. This results in reduced interference between both cameras (Figure 4.3). A different solution to mitigate interference errors is introduced by [Berger et al. 11]. The authors apply a set of fast rotating disks to multiple Kinect^{TM} units, effectively creating a time division multiple access (TDMA). Each disk contains a gap large enough to allow a laser beam to pass through it. Hence, each unit's laser diode is blocked by the disk, except for the time when the gap is allowing the laser to project its pattern into the scene. Each Kinect ^{TM}is equipped with such a disk rotating at the same speed but with a different phase, ensuring that only one laser projects the pattern into the scene at any given time.

4.3 Time-of-Flight Cameras

Technical Foundations In general, there are two main approaches which can be employed in order to realize a ToF camera system without

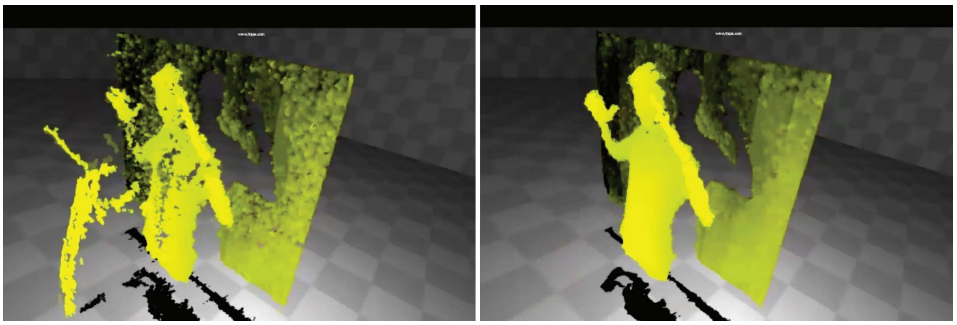


Figure 4.3: Depth-map acquired by a single unit placed within a multi-Kinect setup. Left: significant error in depth due to cross-talk, including depth values being hallucinated. Right: the method in [Butler et al. 12] applied.

requiring expensive coherent light illuminations: continuous wave intensity modulation [Xu et al. 98, Hostica et al. 06, Oggier et al. 05] or pulse-based optical shutter approaches [Iddan and Yahav 01, Yahav et al. 07].

Continuous Wave (CW) Intensity Modulation is the most common approach used in ToF cameras. The general idea is to actively illuminate the scene under observation using near infrared (NIR) intensity-modulated light with a modulation frequency f_m (Figure 4.4). Due to the distance between the camera and the object (sensor and illumination are assumed to be at the same location), and the finite speed of light c, a frequency shift

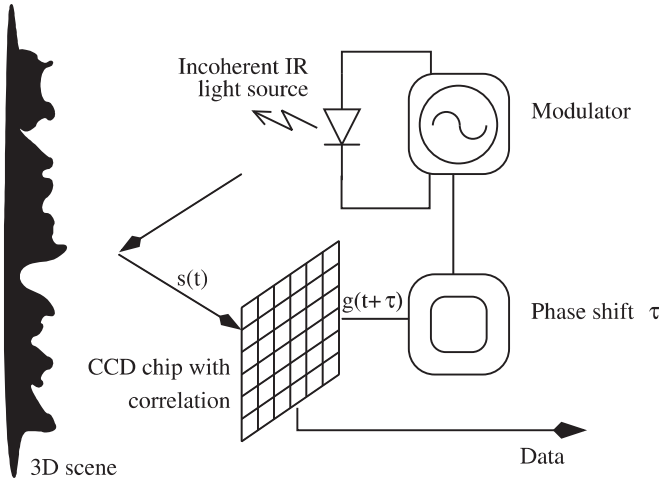


Figure 4.4: The ToF phase-measurement principle.

ϕ is caused in the optical signal which is detected for each sensor pixel by mixing. This information can be easily transformed into the sensor-object distance as the light has to travel the distance twice, i.e.,

$$d = \frac{c\phi}{4\pi f_m}.$$

More technically, the incident optical signal s on each pixel, reflected by the scene, is correlated with the reference generator signal g, possibly with an internal phase offset τ. This appoach, which is also called *mixing*, yields the correlation function which is sampled in each pixel

$$C(\tau) = s \otimes g = \lim_{T\to\infty} \int_{-T/2}^{T/2} s(t) \cdot g(t+\tau)\, \mathrm{d}t.$$

For simple sinusoidal signals $g(t) = \cos(2\pi f_m t)$ the optical signal yields $s(t) = b + a\cos(2\pi f_m t + \phi)$, where a and b are the amplitude and the correlation bias of the incident optical signal, respectively. Some basic trigonometric calculus yields $C(\tau) = \frac{a}{2}\cos(f_m\tau + \phi) + b$.

The demodulation of the correlation function C, i.e., the computation of the phase shift ϕ, is done using several samples of the correlation function C obtained by four sequential phase images with different internal phase offset τ: $A_i = C(i \cdot \frac{\pi}{2}),\ i = 0, \ldots, 3$:

$$\phi = \operatorname{arctan2}(A_3 - A_1, A_0 - A_2), \quad I = \frac{A_0+A_1+A_2+A_3}{4}, \quad a = \frac{\sqrt{(A_3-A_1)^2+(A_0-A_2)^2}}{2}, \tag{4.1}$$

where $\operatorname{arctan2}(y, x)$ is the angle between the positive x-axis and the point given by the coordinates (x, y).

The new Kinect 2™ uses a more complex modulation signal, consisting of various frequencies and different signal shapes. It still applies the basic principle of mixing the optical and the reference signal in order to detect the phase shift.

Figure 4.5 shows recent CW ToF-camera models from different manufacturers. The resolution of these cameras is 120×160 pixel for the CamBoard pico XS and 512 × 424 pixel for the new Kinect™.

Pulse-Based Optical Shutter The pulse-based optical shutter approach is an alternative ToF principle based on the indirect measurement of the time of flight using a fast electronic shutter technique [Iddan and Yahav 01, Yahav et al. 07]. As pulse-based devices are not further discussed, this paragraph only sketches the basic concept of this sensor. The illumination unit sends short NIR light pulses $[t_{\text{start}}, t_{\text{end}}]$ which represent a depth

Figure 4.5: Current ToF cameras: The CamBoard XS (left) from pmdtechnologies and the new Kinect^{TM} 2 from Microsoft (right).

range of interest ("light wall," Figure 4.6). The optical signal is reflected by the scene objects leading to a "distorted" light wall, resembling the objects' shapes. An electronic shutter integrated with the standard CCD chip cuts the front (or rear) portion of the optical signal at the gating time $t_{\text{gate}} = t_{\text{start}} + \Delta_t$; Δ_t resembles the depth offset from the camera to the depth region of interest. The resulting intensity I_{front} is proportional to the distance of the corresponding object's surface. The object's reflectivity and the light attenuation are compensated by using the relation to the completely reflected pulse intensity without gating: $I_{\text{front}}/I_{\text{total}}$. Larger depth ranges may be acquired using several exposures with varying gating parameters [Gvili et al. 03].

Challenges of Time-of-Flight Cameras ToF cameras are active imaging systems that use standard optics to focus the reflected light onto the chip area. Therefore, the typical optical effects like shifted optical centers and lateral distortion need to be corrected for, which can be done using classical intrinsic camera calibration techniques (Section 1.6). However, even though some of the newest ToF cameras achieve almost nearly VGA resolution, most cameras have a resolution in the QQVGA-range or

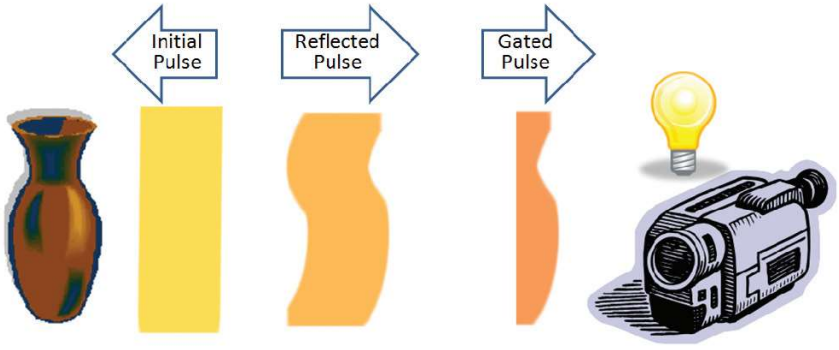


Figure 4.6: The pulse-based optical shutter principle.

even below. This resolution is rather small in comparison to standard RGB- or grayscale-cameras. Thus, for these cameras at or below QQVGA resolution, standard calibration techniques have to be applied with care [Lindner and Kolb 06].

As with any other camera system, ToF cameras can suffer from oversaturation in case of too long exposure times in relation to the ambient background light and the objects' distance and/or reflectivity. Some camera vendors provide a suppression of background intensity on the chip, thus allowing also for outdoor applications. Object areas with extremely low reflectivity or objects far from the sensor lead to a low incident optical signal to the ToF camera, resulting in a bad signal-to-noise-ratio (SNR). Compared to dense stereo, laser scanners and similar approaches, the depth measurement quality is still in the range of several millimeters up to a few centimeters for real-world scenes; thus more accurate reconstructions require spatial data fusion (Chapter 9). Similar to active sensing, the parallel use of several cameras may lead to interference problems, i.e., the active illumination of one camera influences the result of another camera. This kind of interference can be circumvented by using different modulation frequencies.

The ToF principle itself possesses several sensor-specific challenges, which are discussed in detail in the following.

Systematic Distance Error Practically, the generated signal s is not sinusoidal, thus applying the phase reconstruction formulas in Eq.(4.1) yields a systematic error, also called "wiggling" (Figure 4.7 top left). The mean of the systematic error is typically in the order of ± 5 cm after any bias in the distance error has been removed. Furthermore, the systematic error depends on the exposure time of the camera which can be considered as a constant offset with respect to a systematic error at a reference exposure time.

There are several standard approaches to handle the systematic error. First, one has to acquire reference data ("ground truth"). This can be done using track lines [Steitz and Pannekamp 05, Lindner and Kolb 06] or a robot in order to locate the camera in a global reference frame with respect to a known plane [Fuchs and May 08]. Both approaches require rather cost-intensive equipment. Alternatively, vision-based approaches are applied to estimate the extrinsic parameters of the camera with respect to a reference plane, e.g., a checkerboard [Lindner and Kolb 07]. These approaches, however, can only be applied to "high resolution" ToF cameras, additional cameras, or multiple data acquisitions. Correction schemes simply model the depth deviation using a look-up-table [Kahlmann et al. 07] or function fitting, e.g., using b-splines [Lindner and Kolb 06].

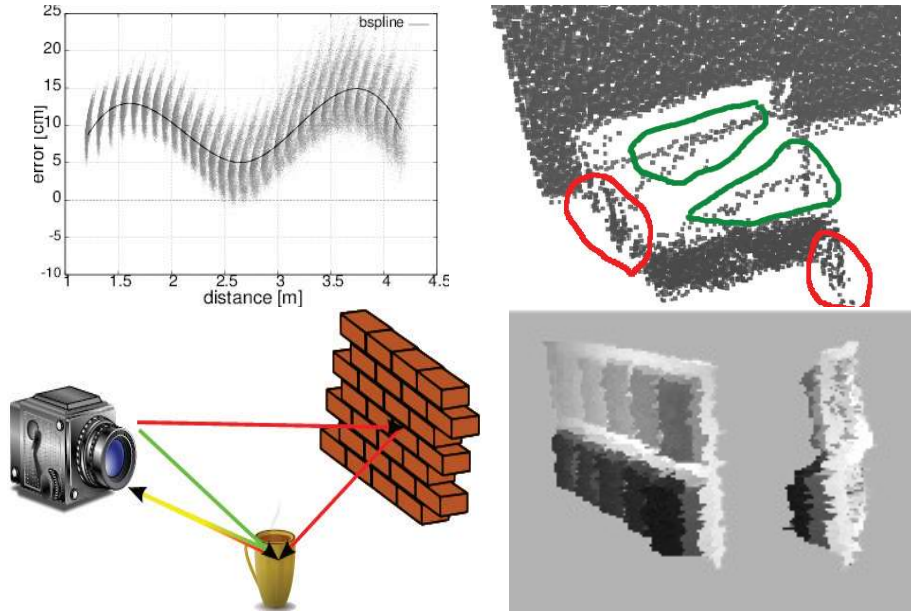


Figure 4.7: Error sources of ToF cameras. Top left: Systematic (wiggling) error for all pixels (gray) and fitted mean deviation (black). Top right: Motion artifacts (red) and flying pixels (green) for a horizontally moving planar object in front of a wall. Bottom left: Schematic illustration of multi-path effects due to reflections in the scene. Bottom right: Acquiring a planar grayscale checkerboard reveals the intensity related distance error.

The systematic error is usually handled in the vendor's device driver. Alternatively, there are open calibration tools like the one from the University of Kiel.[1]

Depth Inhomogeneity At object boundaries, a pixel may observe inhomogeneous depth values. In this case, the mixing process results in a superimposed signal caused by light reflected from different depths, leading to wrong distance values ("flying pixels," Figure 4.7(top right)). There are simple methods relying on geometric models that give good results in identifying flying pixel, e.g., by estimating the depth variance which is extremely high for flying pixel [Sabov and Krüger 10]. Alternatively, more complex approaches such as splitting of pixels, respectively depth image upscaling can be used in combination with gradient methods [Lindner et al. 08] or with additional information from high resolution 2D cameras [Guomundsson and Sveinsson 11].

[1] www.mip.informatik.uni-kiel.de/tiki-index.php?page=Calibration

Multi-Path Effects and Intensity-Related Distance Error *Multi-path effects* relate to an error source common to active measurement systems: The active light may not only travel the direct path from the illumination unit via the object's surface to the detector, but it may additionally travel *indirect paths*, i.e., being scattered by highly reflective objects in the scene or within the lens systems or the housing of the camera itself (Figure 4.7(bottom left)). Within a camera pixel, these multiple responses of the active light are superimposed leading to an altered signal not resembling any meaningful distance information anymore.

One simple approach to compensate for multiple reflections is proposed by [Falie and Buzuloiu 08]. Here the assumption is that the indirect effects are of rather low spatial frequency compared to the direct effects. The authors use differential phase images from neighboring pixels, thus trying to compensate the low-frequency indirect component. Unfortunately, this approach does not work very robustly. Taking only a single pixel into account, it can be shown that any correction scheme requires several modulation frequencies in order to identify superimposed signals in a single pixel. Assuming a perfectly sinusoidal signal, [Dorrington et al. 11] present an analytic formulation for the signal superposition resulting in a highly non-linear optimization scheme which exhibits unstable behavior for specific constellations.

There is another visually obvious artifact which is currently assumed to be a specific form of a multi-path effect, i.e., the *intensity-related distance error*. As expected, darker object regions exhibit a worse SNR resulting in a stronger variation of the depth measurement. The intensity-related error manifests itself as an additional non-zero biased distance offset depending on the amount of incident active light, e.g., resulting from a variation of the objects reflectivity (Figure 4.7(bottom right)).

The specific intensity-related error has mainly been tackled using phenomenological approaches. [Lindner et al. 10] explicitly measure the depth derivation caused by the intensity variation and correct this behavior using a b-spline function fitting.

Dynamic Errors One key assumption of ToF cameras is that each pixel observes a single object point during the acquisition of all phase images. This assumption is violated in case of moving objects or moving cameras, resulting in *motion artifacts*. In real scenes, motion may alter either the observed depth and/or the reflectivity observed by a pixel during the acquisition. Processing the acquired phase images while ignoring the motion present in the acquisition leads to erroneous distance values at object boundaries (Figure 4.7(top right)).

Figure 4.8: Motion compensation: Direct distance computation without compensation (left), and corrected distance (right).

[Schmidt and Jahne 11] detect motion artifacts using temporal gradients of the phase images. In case the gradient in one of the phase images exceeds a certain threshold, motion is detected in this pixel. Correction is performed using extrapolated information from prior frames.

Since motion artifacts result from in-plane motion between subsequent (phase) images, one alternative approach is to use optical flow methods in order to align the individual phase images. [Lindner and Kolb 09] apply a fast optical flow algorithm proposed by [Zach et al. 07] in order to align the three phase images A_1, A_2, A_3 to the first phase image A_0. Optical flow algorithms rely on a brightness consistency assumption which the phase images do not obey, as their "intensity" is the result of a mixing process. The brightness consistency constraint can be fulfilled if the full intensity values for each phase image are available (only few ToF cameras have this option and applying it usually reduces the camera's frame rate due to bandwidth limitations). Additional intensity calibration is applied in order to correct for different gain behavior of pixels and for inhomogeneous

illumation patterns. The optical flow approach is very expensive, resulting in low overall frame rates. [Lefloch et al. 13] propose an optimized approach using only two optical flow computations per depth image (Figure 4.8). A faster approach is to use block-matching techniques applied only to pixels where motion has been detected. They use the direct intensity variation of the phase images which is zero if no motion occurs [Högg et al. 13]. Assuming a linear motion between the phase images and applying a brute force search in the pixel's neighborhood, the flow can be efficiently corrected.

4.4 Summary

Range sensing is one of the longest researched technical challenges in computer vision and photogrammetry. Very recent developments make full-view range depth information available at video frame rates. Fast triangulation-based techniques and time-of-flight cameras offer cheap and easy access to range data at 30 Hz or more. Very current examples are the Kinect^{TM}, and time-of-flight cameras with a focus on continuous-wave approaches.

As mentioned above, the main limitations of current range sensing cameras lie in the limited resolution and the relatively high noise level. The low lateral resolution leads to unwanted effects at object boundaries, e.g., flying or masked-out pixels. As for any active range measuring system, objects with specular or low-reflectivity surfaces cause additional problems, leading to erroneous measurements for both Kinect^{TM}and ToF cameras. For the multi-exposure ToF cameras, temporal errors may occur for fast moving objects and/or cameras.

Despite all the limitations, the development over the last five years is stunning. All the mentioned limitations have been pushed back significantly and it can be expected that there will be further improvements in the upcoming years. These expectations are also backed by the recent release of the second-generation Kinect^{TM}cameras which employ a ToF sensor for depth sensing.

5
Plenoptic Cameras

Bastian Goldlücke, Oliver Klehm, Sven Wanner, and Elmar Eisemann

5.1 Introduction

The light field, as defined by Gershun in 1936 [Gershun 36] describes the radiance traveling in every direction through every point in space. Mathematically, it can be described by a 5D function which is called the *plenoptic function*, in more generality sometimes given with the two additional dimensions time and wavelength. Outside a scene, in the absence of occluders, however, light intensity does not change while traveling along a ray. Thus, the light field of a scene can be parameterized over a surrounding surface; light intensity is attributed to every ray passing through the surface into any direction. This yields the common definition of the light field as a 4D function. In contrast, a single pinhole view of the scene only captures the rays passing through the center of projection, corresponding to a single 2D cut through the light field.

Fortunately, camera sensors have made tremendous progress and nowadays offer extremely high resolutions. For many visual-computing applications, however, spatial resolution is already more than sufficient, while robustness of the results is what really matters. Computational photography explores methods to use the extra resolution in different ways. In particular, it is possible to capture several views of a scene from slightly different directions on a single sensor and thus offer single-shot 4D light field capture. Technically, this capture can be realized by a so-called plenoptic camera, which uses an array of microlenses mounted in front of the sensor [Ng 06]. This type of camera offers interesting opportunities for the design of visual computing algorithms, and it has been predicted that it will play an important role in the consumer market of the future [Levoy 06].

The dense sampling of the light field with view points lying close together may also offer new insights and opportunities to perform 3D reconstruction. Light fields have thus attracted quite a lot of interest in the computer vision community. In particular, there are indications that *small changes* in view point, are important for visual understanding. For

example, it has been shown that even minuscule changes at occlusion boundaries from view point shifts give a powerful perceptional cue for depth [Rucci 08].

5.2 4D Light Field Acquisition

Considering the special case that the light field is recorded on a planar surface, the 4D light field in this sense can be viewed as an intensity function that not only depends on the 2D position on the imaging plane, but also on the 2D incident direction. Many ways to record light fields have been proposed and can be classified into three main categories [Wetzstein et al. 11]. *Multi-sensor capture* solves the problem essentially on the hardware level. One can assemble multiple (video) cameras into a single array, with the cameras lying on a common 2D plane [Wilburn et al. 05]. This solution is quite expensive and requires careful geometric and photometric calibration of the sensors [Vaish et al. 04], as well as considerable effort to process and store the huge amount of data streamed by the array in real-time. However, with temporal synchronization of the camera triggers, one can also apply camera arrays to the recording of dynamic scenes [Wilburn et al. 05]. Furthermore, they allow some interesting applications due to their wide baseline.

In contrast, with *time-sequential imaging* one is limited to static scenes, but only a single sensor is required. Different view points of the scenes are captured consecutively by moving the camera [Levoy and Hanrahan 96, Gortler et al. 96], rotating a planar mirror [Ihrke et al. 08], or programmable aperture, where only parts of the aperture are opened for each shot, allowing to re-assemble the light field from several such images by computational means [Liang et al. 08]. Besides cost considerations, an advantage of the sensor being the same for all views is that calibration is simplified (Chapter 1).

Finally, a technology which recently has become available in commercial cameras is *single-shot multiplexing* where a 4D light field is captured with a single sensor in a single shot, which also makes it possible to record videos. In all cases, one faces a trade-off between resolution in the image ("spatial") and view point ("angular") domain. In plenoptic cameras [Ng 06, Bishop and Favaro 12, Georgiev et al. 11, Perwass and Wietzke 12], spatial multiplexing is realized in a straightforward manner by placing a lenslet array in front of the sensor, which allows to capture several views at the same time. Other techniques include coded aperture imaging [Lanman et al. 08] or, more exotically, a single image of an array of mirrors can be used to create many virtual view points [Manakov et al. 13].

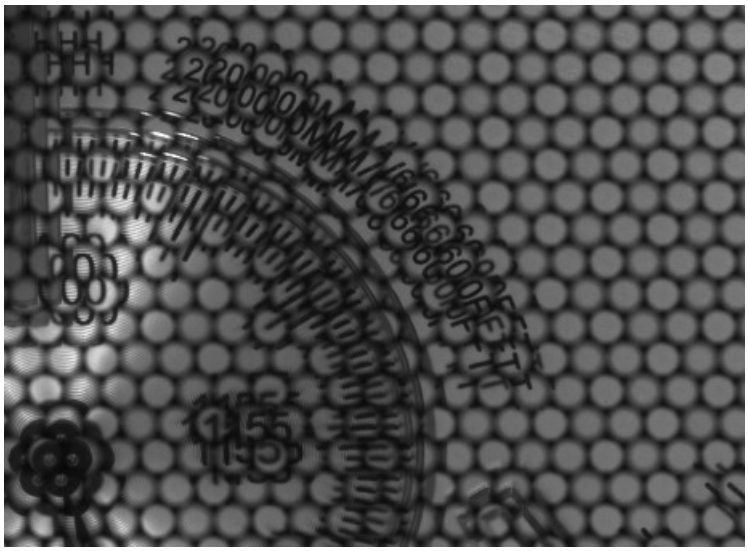


Figure 5.1: Detail of a raw image captured by a plenoptic 2.0 camera by Raytrix. Objects closer to the camera are visible in more microlens images. The camera makes use of different types of microlenses to increase depth of field, which can be distinguished in this image by comparing the sharpness of the projections.

Of the light field acquisition techniques above, plenoptic cameras are gaining increasing interest in the vision community since they are now commercially available as affordable consumer hardware.

5.3 Plenoptic Cameras

While normal 2D cameras only record irradiance from different directions at a single view point in space, plenoptic cameras capture the complete 4D light field on the sensor plane. The idea originates in the early 20th century. First described using a grid of pinholes inside a camera by Ives in 1903 [Ives 03], Lippmann proposed the use of microlenses in front of the image plane in 1908 [Lippmann 08]. Several improvements to the design have been proposed. For example, cameras manufactured by the company Raytrix employ multiple types of microlenses to accomplish a larger depth of field (Figure 5.1).

At the time of writing, plenoptic cameras are commercially available from two manufacturers. The Lytro camera is based on the "plenoptic 1.0" design and targeted at the consumer market, while the Raytrix camera is based on the "plenoptic 2.0" design and targeted at industrial applications. This is reflected in both price as well as the bundled software.

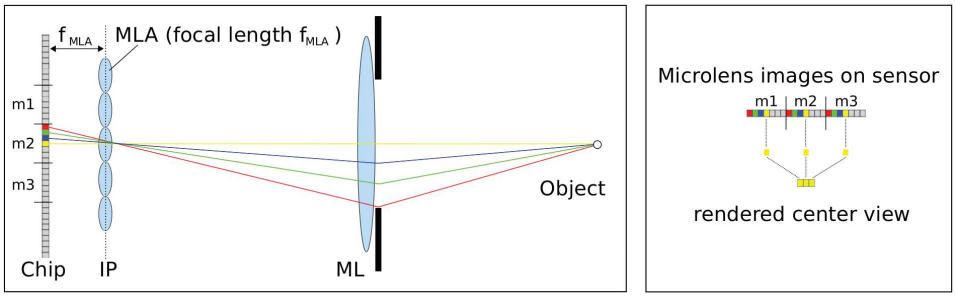


Figure 5.2: *Left:* one-dimensional sketch of a plenoptic camera 1.0 setup. Light rays emitted by the object are focused by the main lens (ML). The microlens array (MLA) is placed at the image plane (IP) of the main lens and thus separates the rays according to their direction. *Right:* a single view point, indexed by (s, t), here the center view, is extracted by collecting the corresponding pixels of each micro image m_i.

The plenoptic camera 1.0 (Lytro camera) design is based on a usual camera with a digital sensor, main optics, and an aperture. In addition, a microlens array is placed in the focal plane of the main lens exactly at the focal length f_{MLA} from the sensor (Figure 5.2). This way, instead of integrating the focused light of the main lens on a single sensor element, the microlenses split the incoming light cone according to the direction of the incoming rays and map them onto the sensor area behind the corresponding microlens. In particular, one has direct access to the radiance $L(u, v, s, t)$ of each ray of the light field by choosing the micro-image of the microlens corresponding to spatial position (s, t) and pixel corresponding to direction (u, v) of the underlying micro-image. The size of each microlens is determined by the aperture or f-number of the main optics. If the microlenses are too small compared to the main aperture, the images of adjacent microlenses overlap. Conversely, sensor area is wasted if the microlenses are too large. Since light passing the main aperture also has to pass a microlens before being focused on a pixel, what actually happens is that the camera integrates over a small 4D volume in light field space. The calibration of unfocused lenslet-based plenoptic cameras like the ones commercially available from Lytro is discussed in [Dansereau et al. 13].

The main disadvantage of the 1.0 design is the poor spatial resolution of the rendered views, which is equal to the number of microlenses. By slightly changing the optical setup, one can increase the spatial resolution dramatically. As another way to compactly record 4D light fields, the focused plenoptic camera has been developed, often called the plenoptic camera 2.0 (Raytrix camera) [Lumsdaine and Georgiev 09, Perwass and Wietzke 12].

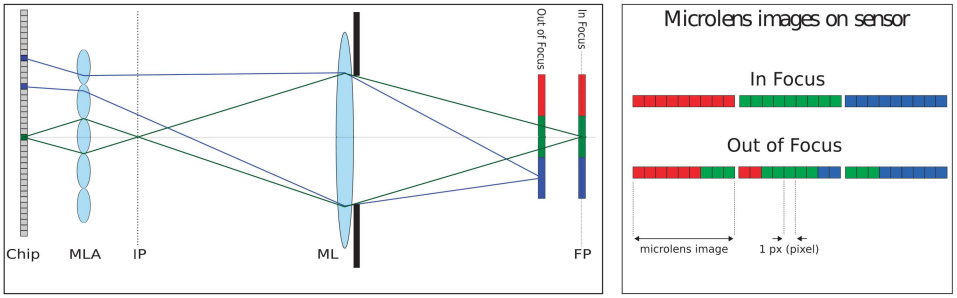


Figure 5.3: *Left:* one-dimensional sketch of a plenoptic camera 2.0 setup. Light rays emitted by the object are focused by the main lens (ML) onto the image plane (IP). The microlens array (MLA) is placed so that the microlenses are focused onto the image plane of the main lens, mapping fractions of the virtual image onto the sensor. Green rays originate from an object in focus of the main lens (FP), blue rays from an object away from the principal plane of the main lens. *Right:* resulting micro-images of an object in and out of focus.

The main difference in the optical setup between the cameras is the relative position of the microlens array. The microlenses are no longer placed at the principal plane of the main lens, but are now focused onto the image plane of the main lens. In effect, each microlens now acts as a single pinhole camera, observing a small part of the virtual image inside the camera. This small part is then mapped with high spatial resolution onto the sensor. The scene points have to lie in a valid region between the principal plane of the main lens and the image sensor. Scene features behind the principal plane cannot be resolved.

Scene points that are not in focus of the main lens but within this valid region are imaged multiple times over several neighboring microlenses, thus encoding the angular information over several micro-images (Figure 5.3 [Lumsdaine and Georgiev 09, Perwass and Wietzke 12]). Angular information is encoded while at the same time preserving high resolution. Due to multiple imaging of scene features, however, rendered images from this camera have a lower resolution than the inherent sensor resolution promises. The light field is encoded in a complicated way, and it is necessary to perform an initial depth estimate at least for each microlens in order to decode the sensor information into the standard 4D light field data structure [Wanner et al. 11]. External and internal calibration of plenoptic 2.0 cameras has been investigated in [Johannsen et al. 13].

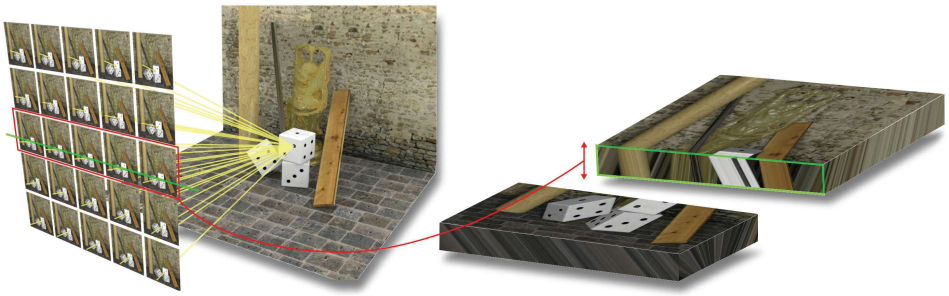


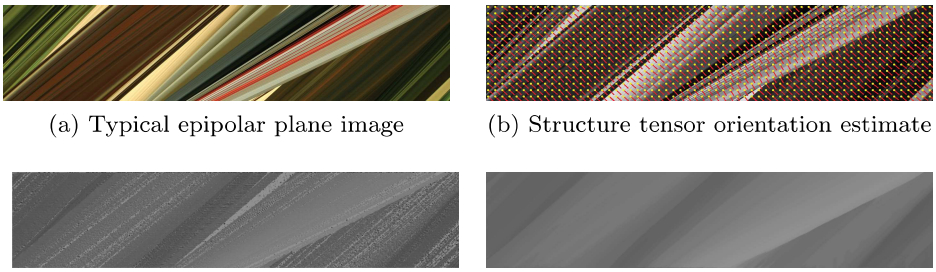
Figure 5.4: One way to visualize a 4D light field is to think of it as a collection of images of a scene, where the focal points of the cameras lie in a 2D plane. The rich structure becomes visible when one stacks all images along a line of viewpoints on top of each other and considers a cut through this stack (denoted by the green border). The 2D image one obtains in the plane of the cut is called an *epipolar plane image (EPI)*.

5.4 4D Light Field Structure and Depth Reconstruction

Since a 4D light field can be understood as a dense collection of multiple views, off-the-shelf correspondence search techniques can be applied to infer 3D structure (Chapter 8). Due to the rich information content in the light field data, however, also specialized methods can be developed, which work more efficiently and robustly.

One line of research follows the philosophy of the earliest works on the analysis of epipolar volumes [Bolles et al. 87], and rely on the fact that 3D scene points project to lines in the epipolar-plane images. The reason is that a linear camera motion leads to a linear change in projected coordinates (Figure 5.4). These lines can be more robustly detected than point correspondences which have been exploited in several previous works [Bolles et al. 87, Berent and Dragotti 06, Criminisi et al. 05]. A recent advanced method aims at accurate detection of object boundaries and is embedded in a fine-to-coarse approach, delivering excellent results on very high-resolution light fields [Kim et al. 13].

In the same spirit, an efficient and accurate approach, which is however limited to only small disparity values and thus has a limited depth range, computes a direct estimate of the local orientation of the pattern [Wanner and Goldlücke 14] (Figure 5.5). Here, orientation estimation is performed using an eigenvector analysis of the first-order structure tensor of the EPI. This approach can be extended to detect multiple overlaid



(a) Typical epipolar plane image (b) Structure tensor orientation estimate

(c) Noisy local depth estimate (d) Depth estimate after optimization

Figure 5.5: Depth estimation on an epipolar plane image (a). Standard 2D pattern analysis using the structure tensor yields a robust orientation estimate (b), whose slope encodes the (still noisy) depth map for the EPI (c). Global optimization techniques result in a consistent estimate across all views (d).

patterns to efficiently reconstruct reflections or transparent objects [Wanner and Goldlücke 13]. Since local depth estimates from any source (including, e.g., stereo matching) are usually noisy, global smoothing schemes can be employed to improve the result. By careful construction of the regularizers and constraints, one can obtain consistent estimates over the complete light field which respect occlusion ordering across all views [Goldlücke and Wanner 13, Wanner and Goldlücke 14].

Other 3D reconstruction methods specific to light fields exist, including focus stacks in combination with depth-from-focus methods [Nayar and Nakagawa 94, Perez and Luke 09]. Multiple methods that make use of depth maps to warp individual light field views to densify the light field from a sparse set of views have been proposed (Section 17.2).

5.5 Spatial and Angular Super-Resolution

Since plenoptic cameras trade off sensor resolution for the acquisition of multiple view points, it is not surprising that super-resolution techniques are one focus of research in light-field analysis. Such methods have been investigated using priors regarding statistics of natural images [Bishop and Favaro 12] as well as modified imaging hardware [Lumsdaine and Georgiev 09].

In the classical Bayesian approach, an image formation model is set up to obtain the known input images from the desired super-resolved target image. In particular, when one transforms the target image into the image domain of an input image and performs a downsampling operation (usually

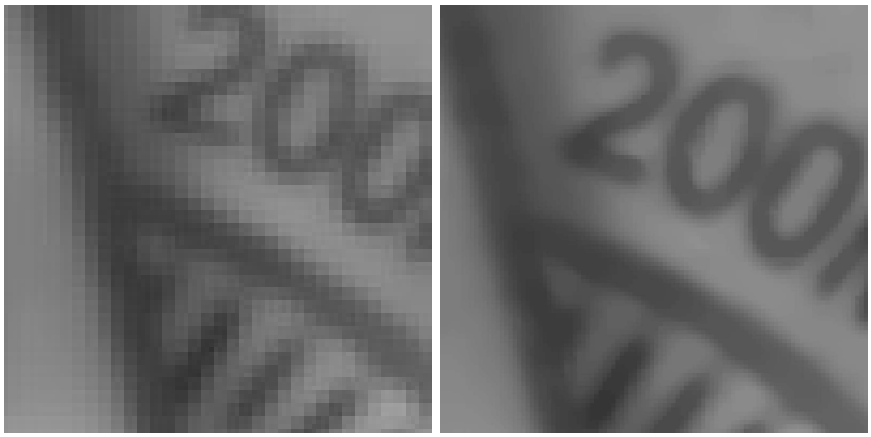


Figure 5.6: By solving a single inverse problem, one can create super-resolved novel views from a 4D light field captured with a Raytrix plenoptic camera [Wanner and Goldlücke 12]. Above are close-ups of one of the 7×7 input views (left) and the result from the super-resolution algorithm (right).

modeled via a blur kernel), one should obtain an exact copy of the input image. In practice, however, this property will not be satisfied exactly due to sensor noise or sampling errors. Thus, the set of equations is enforced as a soft constraint in a minimization framework, where the desired super-resolved image appears as the minimizer of some energy functional [Bishop and Favaro 12, Wanner and Goldlücke 12].

In particular, some frameworks also allow to generate views in new locations, thus solving an image-based rendering task in the same step [Wanner and Goldlücke 12]. In some recent work, a Bayesian framework was explored which also models uncertainties in the depth estimates and which is able to mathematically derive many of the heuristics commonly used in image-based rendering [Pujades et al. 14]. The topic of image-based rendering is explored in detail in Chapter 17.

5.6 Refocusing and Other Applications

In this section, methods are presented that allow to simulate the intrinsics of a usual camera by relying on a light field as input. The two additional dimensions of a 4D light field compared to a conventional 2D image (quantities: radiance $[\mathrm{W\,m^{-2}\,sr^{-1}}]$ vs. irradiance $[\mathrm{W\,m^{-2}}]$) make it possible to produce effects such as changing the aperture or refocusing (adjusting the focal plane) even *after* a photo has been taken.

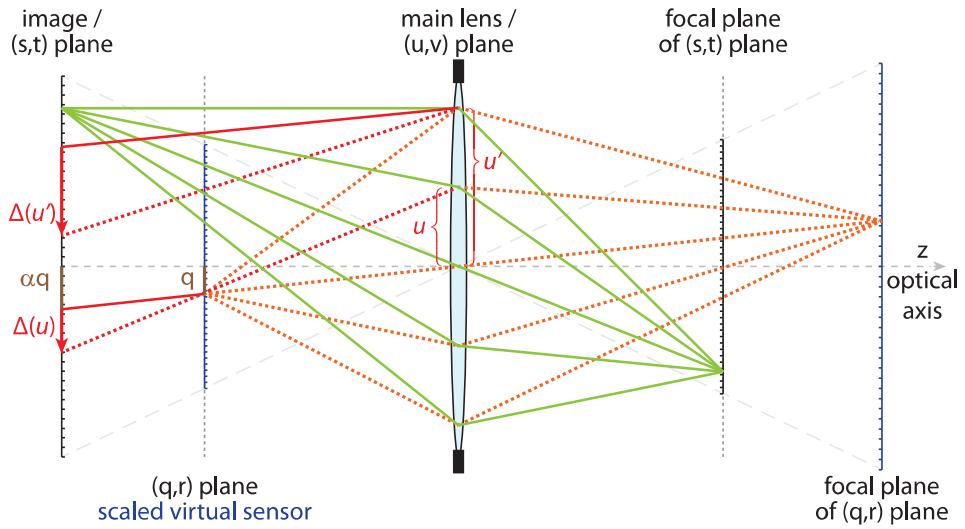


Figure 5.7: Refocusing example: a new virtual sensor at plane (q, r) is introduced, causing a different focal plane. The mapping from coordinates (q, r) in the local space of the scaled sensor to coordinates (s, t) in the space of the original image plane reduces to a constant translation $\Delta(u, v)$.

For many of these effects, a depth image has to be computed first, which can be directly derived from the light field (Section 5.4). In particular, the plenoptic camera 2.0 requires a reasonable depth estimate to reconstruct any meaningful image from the captured light field. This depth reconstruction is possible because the light field stores partially redundant information. More precisely, objects in a scene tend to have similar appearance under slightly different viewing angles. While the redundancy can be directly used for compression of light field data [Levoy and Hanrahan 96], it has recently been exploited for the reconstruction of a light field from very sparse data [Marwah et al. 13].

The basis of the following examples is to sample or integrate the 4D light field to synthesize a new 2D image. The classical (u, v, s, t) parameterization [Levoy and Hanrahan 96] of a light field uses two distinctive planes that are aligned with the optical axis z: $[u, v]^T$ denotes the coordinates on the plane at the main lens (ML) z_{UV} and $[s, t]^T$ the coordinates on the focal plane of the main lens. As points on the focal plane uniquely map to points on the image plane (IP), $[s, t]^T$ also denotes the coordinates on the image plane at z_{ST}.

The light field can be used to fetch radiance for a new plane (q, r) at distance z_{QR}, parallel to the (s, t) plane. The mapping to the original coordinates is simple as it only requires to find the intersection of the ray,

Figure 5.8: Example of a refocusing sequence. Left to right: the focal plane is moved from front to back, shifting the focus from the buddha statue to the pirate. The scene was captured with the kaleidoscope camera add-on [Manakov et al. 13] with an 50mm f/1.4 main lens. While this light-field camera add-on only captures nine directions, these views are sufficient to estimate depth and perform view interpolation, allowing for smooth out-of-focus blur.

originating at $[q, r, z_{\mathrm{QR}}]^T$ with direction $[u, v, z_{\mathrm{UV}}]^T - [q, r, z_{\mathrm{QR}}]^T$ with the (s, t) plane at z_{ST}. The $[s, t]^T$ coordinates of the intersection point can be determined in two steps: first, a scaling α of $[q, r]^T$ depending on the positions of the (s, t), (q, r), and (u, v) planes is computed: $\alpha = \frac{z_{\mathrm{UV}} - z_{\mathrm{ST}}}{z_{\mathrm{UV}} - z_{\mathrm{QR}}}$. Second, a translation by $\Delta(u, v) = -\beta \cdot [u, v]^T$ with $\beta = \frac{z_{\mathrm{QR}} - z_{\mathrm{ST}}}{z_{\mathrm{UV}} - z_{\mathrm{QR}}}$ yields the final coordinates in the (s, t) plane: $[s, t]^T = \alpha \cdot [q, r]^T + \Delta(u, v)$ (Figure 5.7).

While a pinhole camera could, in theory, have an infinitesimal aperture, such a camera would not produce any image, because no light would be detected. Hence, cameras rely on a larger aperture and use a lens to refocus the rays. All points on a so-called focal plane project to exactly one location on the sensor; outside the focal plane, points can project to several locations. Adjusting the focal plane right is a major challenge in photography. Imagining light rays leaving a camera, all rays from a given pixel will meet on the focal plane. Traversing these light rays in the opposite direction, all rays will be integrated at the given sensor pixel. With 4D light fields, it is possible to perform this integration in a post-capture process (Figure 5.8).

A usual camera with the sensor at the image plane (IP) is simulated by integrating over all directions, hence, the (u, v) plane. Roughly, for a plenoptic camera 1.0, all pixels under a microlens are summed up as: $L(s, t) := \sum_u \sum_v L(u, v, s, t)$. The focal plane depends on the distance of the IP to the ML. Assuming the thin lens model, the original focal plane is at a distance $d_{\mathrm{org}} = (1/f - 1/(z_{\mathrm{UV}} - z_{\mathrm{ST}}))^{-1}$ from the main lens, where f is the focal length of the main lens. A virtual move of the image plane to a (q, r)-plane at z_{QR} causes the focal plane to change. Precisely, the new focal plane will be located at a distance $d_{\mathrm{refocus}} = (1/f - 1/(z_{\mathrm{UV}} - z_{\mathrm{QR}}))^{-1}$. To evaluate the result with the new focus plane, from a point on (q, r) all

rays toward (u, v) are integrated, whereby (u, v, q, r) is mapped to (u, v, s, t) coordinates by the ray/plane intersection method described above.

This approach can be rendered more efficiently by splatting individual views (each indexed by their (u, v) coordinates (Figure 5.2 right)). Entire scaled views indexed by (u, v) can be accumulated on the sensor: $L(q, r) = \sum_u \sum_v L(\alpha q + \Delta(u), \alpha r + \Delta(v), u, v)$ with $\Delta(u), \Delta(v)$ denoting the first respectively second component of Δ.

The main challenge of refocusing is that it requires a very high number of different view points in order to achieve a smooth out-of-focus blur. For a large blur kernel, banding or ghosting artifacts can remain visible. As none of the existing plenoptic cameras provides a sufficiently high number of view points, it is often essential to perform view interpolation (Sections 5.5 and 17.2).

In photography, *Bokeh* defines the rendering of out-of-focus areas by a camera lens. For small and very bright out-of-focus lights, this effect can be strong and is used as a stylization method. The shape of the Bokeh is indirectly defined by the shape of the lens aperture. As the lens aperture in a standard camera cannot be changed, photographers often attach an additional aperture with reduced size in front of the lens. The attachment simply blocks incoming light from certain directions. With the 4D light field, it is very simple to simulate such a behavior: $L'(u, v, s, t) = b(u, v)L(u, v, s, t)$ with b being a function mimicking the aperture shape. In order to control the aperture, incident light rays are thus scaled by a weighing factor (usually a binary mask). Additionally, it is possible to make this influence depend on the wavelength.

As refocusing practically requires interpolation in the (u, v) domain to generate additional views (Sections 5.4 and 5.5), the same pipeline can also perform extrapolation. Extending the available directional domain corresponds to photography with a larger aperture, which allows for very narrow depth-of-fields. Manakov et al. [Manakov et al. 13] report the simulation of a lens with an aperture of up to $f/0.7$ from a single snapshot light-field.

In practice, pixels of a light-field camera do not correspond to exact rays. Instead, each pixel records the incident irradiance within a small cone of directions. Each view point that relates to the microlens corresponds to an image taken with a lens of small aperture (Figure 5.2 right). Consequently, these views also exhibit depth-of-field, and any refocusing operation is limited to the depth-of-field range imposed by these optics. Similarly, the captured light field might not be sufficient to deal with large apertures as some light rays necessary for the border pixels might be missing.

In a 4D light field, when keeping (s, t) constant, varying the (u, v) parameters results in a view of the scene (Figure 5.2 right), which roughly corresponds to a capture of the scene with a pinhole camera centered at (u, v). Changing the (u, v) parameters causes a "lens-walk" and offsets the

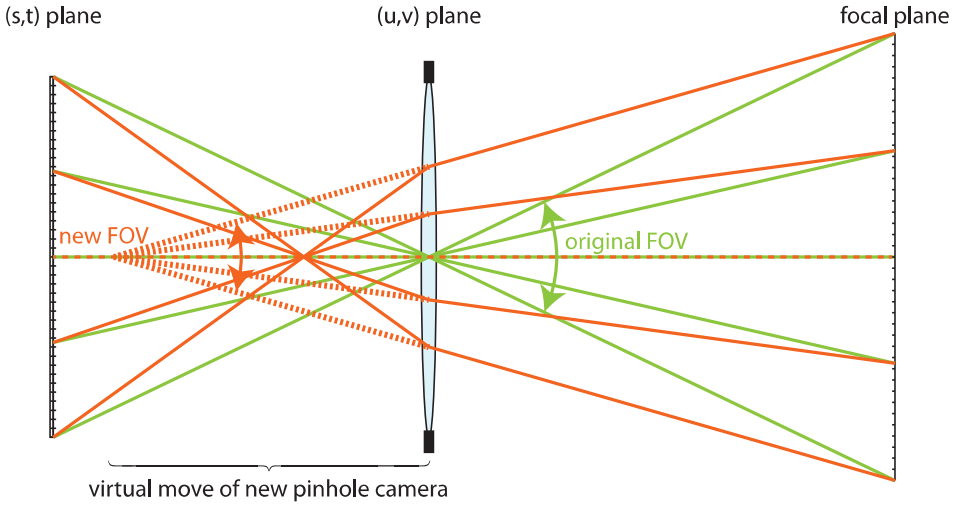


Figure 5.9: Dolly zoom: the light field allows to pick a (u, v) per (s, t). While the center view (green) corresponds to a pinhole camera with center of projection at the origin of the (u, v) plane, it is possible to simulate a moving pinhole camera with different field-of-view. Here, objects in front of the focal plane shrink while objects behind the focal plane grow in projected size (and are potentially cut off).

corresponding image. This 2D effect relates to the Ken Burns-effect using a 2D pan and zoom, but with a light field this walk can also be extended to 3D by varying the (s, t) coordinates. Hereby, a parallax effect is induced due to the different view points. A trivial extension is the generation of stereo images by sampling two (u, v) views. More details on multi-view-stereo are described in Section 8.3.

A computationally more involved effect is the "dolly zoom" or "Hitchcock zoom," where a change of the field-of-view (FOV) and camera motion along the viewing direction are synchronized, while focusing on an object in the scene. It causes out-of-focus regions to warp due to the changing FOV while the focal plane position and its imaged extent remain the same. Typically, this effect is used to shrink/grow the background, while keeping the object in focus at the same scale for a dramatic effect. To achieve this result, the image is rendered by: $L(s, t) = L(s, t, \gamma s, \gamma t)$ with γ defining the strength and direction of the effect. Here, a single ray sample is taken from each view (Figure 5.9).

While refocusing processes thousands of views for high-quality rendering, the dolly zoom requires a single view per pixel. In turn, the computational complexity stems from the fact that it requires dense directional information. In practice, angular interpolation is strictly needed. While

splatting of entire views, as in the refocusing application, is not possible, some computational simplifications can be made. The effect is most efficiently implemented by coupling the ray selection $(s, t, \gamma s, \gamma t)$ and directional interpolation in a GPU shader.

5.7 Summary

With the advent of consumer-grade plenoptic cameras, light field imaging has become comparatively cheap. Acquisition of a 4D light field is now as simple as taking a picture with a standard digital camera. Consequently, in addition to the traditional light field applications in computational photography and image-based rendering, a lot of research interest has been geared recently to leverage light fields for computer vision challenges like non-Lambertian 3D reconstruction.

Related to this chapter, Chapter 8 deals with 3D reconstruction from light field correspondence estimation, while Chapter 17 covers image-based rendering in more detail.

6
Illumination and Light Transport

Martin Fuchs and Hendrik P.A. Lensch

6.1 Introduction

Correctly modeling scene illumination is crucial for many applications in real-world visual computing. Many applications require painstaking control and repeatable conditions—for instance, whenever illumination response is used to infer scene geometry, such as in photometric stereo or shape-from-shading. Other applications can abstract from illumination, once it is precisely known. Some methods enable editing the effect of incident illumination on a scene, a technique known as *relighting*, in a way which takes into account both direct reflections, especially diffuse color and specular highlights, as well as more complex indirect effects such as interreflections, subsurface scattering or caustics.

This chapter is concerned with the last group of techniques. It deals with the problem of digitally recording illumination and its effects on scene appearance. Recently, Heide et al. and Velten et al. have proposed sophisticated optical implementations which are able to temporally resolve light transport and create visualizations of light propagating through a scene [Heide et al. 13, Velten et al. 13]. In contrast, this chapter addresses the cumulative effect of illumination on a scene, especially the connection between incoming and observed light as mediated by global light transport.

Exhaustive recording of light transport will likely remain intractable for the foreseeable future. Accordingly, the design of acquisition setups is intrinsically intertwined with choosing an underlying, possibly simplifying, light transport model; as a result, this chapter discusses global modeling of light transport before addressing practical issues of recording.

6.2 Modeling Illumination

Illumination can be exhaustively expressed as a spectral radiance distribution of a 4D incident light field (Chapter 5). Many visual computing

applications, however, make the simplifying assumption that illumination is *distant* from the scene being recorded, that is, it is so far away that its variation over the scene surface points is negligible, and hence, in essence, it changes only with incident direction.

In this case, measured illumination can be stored in a 2D data structure, the *environment map* [Blinn and Newell 76]. Environment maps may be parameterized in a variety of ways, the choice of which enables multiple trade-offs regarding *storage efficiency* (parameterizations which fill a single or at least a small number of rectangular texture without wasting space are advantageous here), *rendering efficiency* (parameterizations which can be efficiently sampled, and/or permit a rasterization pipeline to compute the environment map are useful), and finally *low distortion* regarding straight lines and variation of the ratio between solid angle and mapped area. The latter needs to be considered especially when an environment map is used for image-based illumination in rendering, as distortions in solid angle need to be numerically compensated when integrating over the incident illumination. Unfortunately, these trade-offs can generally not be optimized simultaneously.

Aside from distant illumination, other modeling assumptions may also enable storage as a flat image: in cases where illumination can be expressed as a projective mapping between a plane and a ray distribution—as would be the case for digital projectors, as long as the pinhole model is applicable—the parameterization as coordinates in this plane arises naturally as a reasonable choice.

6.3 Measuring Illumination

It is possible to record an environment map by using a camera with a very wide field-of-view (*"fish-eye lens"*), or perform panoramic stitching of several images of a single camera, which is rotated by its optical center (Chapter 3). For recording environment maps special camera rigs, such as the PointGrey Ladybug or the Google Street View cameras have been developed. Spheron offers high-resolution HDR cameras with a rotating line sensor. A more common, cheaper approach, instead, is to record an HDR image of a *light probe*, an object of known geometry and reflectance, and infer the incident illumination from its appearance.

The most common light probe design consists of a ball shape with a mirroring surface material [Debevec 98] (Figure 6.1). This serves two purposes: the rotationally symmetric geometry of a sphere makes registration, that is, the mapping between camera pixels and observed surface geometry, easier, as merely the silhouette of the sphere needs to be detected in a recorded picture. The mirror surface enables a simple environment map

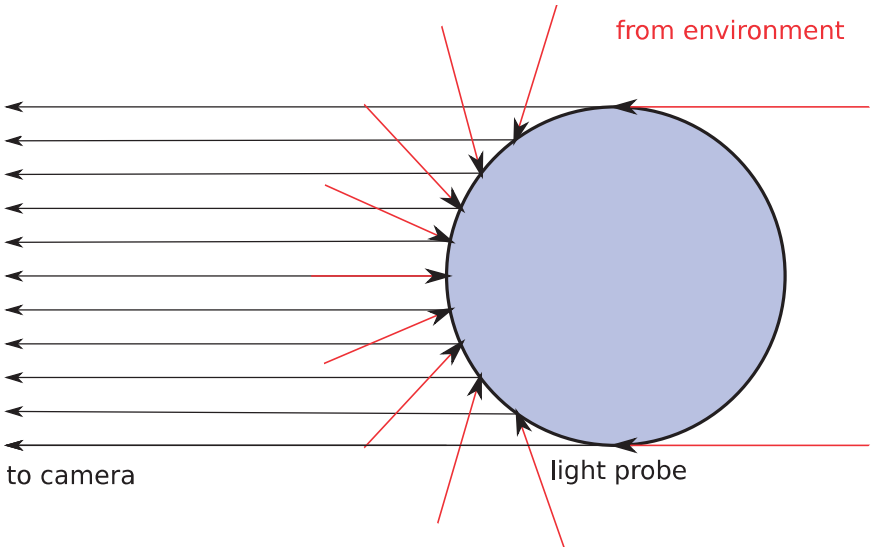


Figure 6.1: A spherical light probe ideally provides a $360° \times 180°$ view of the environment to an orthographic camera. As this illustration of light ray distribution shows, equidistant sampling on the camera creates parameterization problems at the silhouette of the sphere, reducing the usability of grazing-angle observations.

recovery scheme: any ray casting implementation may be used to trace rays through the camera pixels, reflect them at the surface of the estimated sphere, and deposit the image information in the pixel value as radiance measurement in the output environment map. There, the found values need to be interpolated to fill eventual gaps.

In an ideal recording scenario with an orthographic camera, a single image of the sphere reveals a full $360° \times 180°$ environment map. In practice, even though a long focal length can be used to approximate an orthographic projection, a blend between two recordings taken from opposite directions is preferable: environment map directions behind the light probe are observed in pixels close to the sphere silhouette where mis-registration of the sphere position has the largest effect and where the sampling density between orthogonal directions is most unevenly distributed. Additional recording positions can be useful in order to avoid the central pixels which show the recording camera being reflected on the sphere.

Even putting aside specifically crafted optical implements, several options for procuring a mirror ball are available, the least expensive being a holiday ornament such as a Christmas bauble. However, while the

manufacturing techniques result in a high-quality surface finish, the material shape is typically so uneven that a precise measurement of the geometry in combination with elaborate registration of the sphere orientation may be required. More precise geometry may be found in ball bearing balls, the surface of which, however, has little resistance to scratching and which are only widely available in limited diameters up to a few centimeters.

Fuchs et al. use a black snooker ball as a light probe, which has both a smooth surface, sharp highlights, and is comparably resistant mechanically [Fuchs et al. 05]. However, due to its non-metallic material, the pixel values cannot be used directly to reconstruct an environment map.

More precise HDR estimates can be obtained by combining a mirror, a dark specular, and a diffuse sphere [Stumpfel et al. 04] to recover the environment map, very bright light sources, and the correct total irradiance. For outdoor scenarios, a low-parameter sky and sun illumination model can be indirectly recovered from a single image without any specific light probe [Lalonde et al. 09].

6.4 Modeling Light Transport

In contrast to just capturing the incident illumination in the form of an environment map, the light transport in an arbitrary scene is a complex, non-local phenomenon which for opaque surfaces in free space is governed by the *rendering equation*

$$L(\mathbf{x}, \boldsymbol{\omega}_o) = L_e(\mathbf{x}, \boldsymbol{\omega}_o) + \int_\Omega f_r(\boldsymbol{\omega}_o, \mathbf{x}, \boldsymbol{\omega}_i) L(\mathbf{x}, \boldsymbol{\omega}_i) \cos\theta \, d\boldsymbol{\omega}. \tag{6.1}$$

The radiance $L(\mathbf{x}, \boldsymbol{\omega}_o)$ leaving point $\mathbf{x}$ in direction $\boldsymbol{\omega}_o$ combines the self emission L_e and the light L incident from all possible directions Ω that is scattered at $\mathbf{x}$ toward $\boldsymbol{\omega}_o$ by the *bidirectional reflectance distribution function (BRDF)* f_r. Here, the light transport can become quite complex as the light reflected at one scene point might indirectly illuminate other scene points due to interreflections, causing so-called global effects. In scenes with participating media and transparent or translucent materials even more complex interaction will occur.

Reasoning about light transport becomes significantly easier when writing the rendering equation in operator notation

$$\mathbf{L} = \mathbf{L}_e + \mathbf{K}\mathbf{L}. \tag{6.2}$$

The process of scattering and reflection typically acts linearly on the light field $\mathbf{L}$ and the transformation of the light field due to single scattering events can be modeled by the linear operator $\mathbf{K}$, effectively representing

the convolution of the incident light field with the BRDF. Global light transport thus is the solution of Eq.(6.2):

$$\mathbf{L} = (\mathbf{I} - \mathbf{K})^{-1} \mathbf{L}_e \tag{6.3}$$

When measuring the light transport in a scene one typically can only measure a subset $\mathbf{C} \subseteq \mathbf{L}$ of the entire light field, typically a single image or a collection of images. Denoting the present illumination $\mathbf{L}_e$ by $\mathbf{L}$ and the transport operator $\mathbf{T} = (\mathbf{I} - \mathbf{K})^{-1}$, most measurement approaches are governed by a simple linear equation:

$$\mathbf{C} = \mathbf{TL} \tag{6.4}$$

Provided some controlled illumination $\mathbf{L}$ and the measurements $\mathbf{C}$, the task is to identify the operator $\mathbf{T}$. Knowing $\mathbf{T}$ allows for synthesizing novel images $\mathbf{C}$ under arbitrary illumination, a process called *relighting*.

The transport operator $\mathbf{T}$ often is also referred to as a *reflectance field* R [Debevec et al. 00] that maps the incident light field to the reflected light field incorporating all global light transport effects. Written as an integration over all incoming directions $\boldsymbol{\omega}_i \in \Omega$ and scene points $\mathbf{x}_i \in S$,

$$C(\mathbf{x}_o, \boldsymbol{\omega}_o) = \int_\Omega \int_S R(\mathbf{x}_i, \boldsymbol{\omega}_i, \mathbf{x}_o, \boldsymbol{\omega}_o) L(\mathbf{x}_i, \boldsymbol{\omega}_i)\, ds\, d\boldsymbol{\omega} \tag{6.5}$$

Insights on the particular structure of $\mathbf{T}$ can be found in [Veach 97, Garg et al. 06, Ramamoorthi and Hanrahan 01c, Seitz et al. 05]: Veach [Veach 97] analyzes the structures that appear in the context of rendering synthetic scenes. Garg et al. [Garg et al. 06] show that direct reflections cause a sparse manifold in $\mathbf{T}$ while global effects tend to introduce partially dense interaction in $\mathbf{T}$ albeit typically with rather low-rank structure. Ramamoorthi and Hanrahan [Ramamoorthi and Hanrahan 01c] analyze the frequency response of light transport from first principles and derive resulting bandwidth constraints when recovering reflectance or illumination. The invertibility of the operator given partial information has been investigated by Seitz et al. [Seitz et al. 05].

One important property of the light transport operator is that it incorporates Helmholtz reciprocity: in the same optical medium, the reflectance observed along a path will be the same as if observed along the same path just in the opposite direction. Thus, with the right setup one measurement can provide information from both sides. This can be used for stabilizing or accelerating the measurement process in the form of dual photography approaches [Sen et al. 05, Garg et al. 06, O'Toole et al. 12].

Techniques for both relighting or light transport measurement can be categorized based on the way the transport operator $\mathbf{T}$ is represented or identified.

By discretizing the light field or camera image as well as the incident illumination, the transport operator maps from illumination to pixel basis functions. For example, the complete reflectance field between a one megapixel camera and the illumination from a one megapixel environment map already yields a matrix $\mathbf{T}$ of $1,000,000 \times 1,000,000$ entries. As the size of $\mathbf{T}$ quickly becomes impractical, many variants have been proposed.

Some approaches directly sample $\mathbf{T}$ as the impulse response to a discrete set of illumination stimuli. Most often, point light sources (or projector rays) are turned on in sequence each time capturing one image (or light field) corresponding to one column of the transport matrix ([Debevec et al. 00, Goesele et al. 04, Fuchs et al. 07]). Relighting then is achieved by vector-matrix multiplication, i.e., each input image is scaled by the desired light source color for that particular direction and the relit image is obtained by summing up all scaled images.

Modeling the illumination in a lower-dimensional space with correspondingly fewer basis functions, e.g., spherical harmonics, the number of required measurements can be drastically reduced [Ghosh et al. 10, Tunwattanapong et al. 13] by measuring the system response to each basis function.

Adaptive schemes try to identify regions in the reflectance function that require sampling with higher resolution compared to others. For example, when the observed reflectance at a novel sampling location can already be explained by interpolation of the coarser samples, the local resolution is already sufficient [Fuchs et al. 07]. Closely related to adaptive schemes is the representation of $\mathbf{T}$ by hierarchical bases. Approaches use, for example, wavelets [Peers and Dutré 03, Sen et al. 05, Peers et al. 09] or hierarchical, low-rank approximations such as H-matrices [Hackbusch 99, Garg et al. 06].

Further simplifications are possible by constraining the space of possible transport matrices. In the simplest form, one focuses on direct reflections only. By identifying only the peak in each row one can derive a simple scheme to project an image onto an arbitrary surface. Inverting the direct component, one can efficiently reduce the influence of the spatially varying reflectance of the surface [Wetzstein et al. 07]. In environment matting [Zongker et al. 99, Matusik et al. 02], the contribution to each camera pixel is limited to a Gaussian distribution around the peak transport coefficient, allowing to approximate effects such as blurred refraction or reflection, but not to truly express global light transport.

Related to the problem of measuring global transport is the measurement of local appearance where the appearance of a specific surface or 3D object is to be captured. Spatially varying reflectance then is parameterized by the surface and the appearance for each surface point modeled by some surface reflection model. Weyrich et al. provide a concise overview of these techniques [Weyrich et al. 08].

Note that the equations given are expressed for total quantities of radiance but can just as well be interpreted as referring to wavelength bands of spectral radiance or color channels. Wavelength-changing effects can be incorporated by an additional dimension to integrate over and expressing distributions over incoming and scattered wavelength.

6.5 Measuring Light Transport

The problem of actually acquiring light transport is closely related to the chosen representation of both the reflectance field and the incident illumination.

Choice of Illumination

Early research focused on point light sources providing highly controllable illumination. The positions of the light sources here are either fixed, as in a light stage [Debevec et al. 00], manually or adaptively positioned [Masselus et al. 02, Lensch et al. 03, Fuchs et al. 07] or treated as unknowns to be recovered in the reflectance estimation process. In order to achieve denser sampling of the incident hemisphere, monitors may be used as light sources. They provide addressable resolution in the millions of pixels, but, as a consequence, a direct sampling approach is practically intractable. Instead, hierarchical schemes are employed [Zongker et al. 99, Peers and Dutré 05, Sen et al. 05].

The use of projectors [Masselus et al. 03, Goesele et al. 04, Sen et al. 05, Garg et al. 06] allows for capturing both directional dependency as well as localized spatial scattering phenomena, but at the same time it increases measurement complexity as 4D illuminations are used.

Reconstruction Algorithms

As the cost of brute-force sampling of all light sources quickly becomes too expensive, some algorithms explore the structure of the light transport, reconstructing the constrained operator from fewer measurements rather than measuring it directly.

Using varying natural illumination conditions and capturing the actual incident illumination with a light probe, Matusik et al. reconstruct the reflectance field for outdoor relighting by least squares minimization [Matusik et al. 04]. Fuchs et al. capture indoor scenes under varying illumination, using the incident environment map implicitly as an illumination basis [Fuchs et al. 05]. For relighting, a Bayesian framework projects the intended illumination into the basis of captured illuminations, producing a set of weights that are used to blend the input images directly.

Based on adaptive, hierarchically refined illumination patterns, Garg et al. recover a block-wise rank-1 approximation of the light transport based in H-matrices [Garg et al. 06]. Blocks and the corresponding illumination patterns get refined as long as the rank-1 approximation does not hold.

General low-rank approximations of the overall light transport can be recovered using the Kernel–Nyström method by successively exploring a higher-rank reconstruction with each additional measurement [Wang et al. 09]. O'Toole and Kutulakos directly measure the eigenvectors of the light transport operator by computing the next illumination pattern using the Krylov subspace method [O'Toole and Kutulakos 10]. By illuminating with the next illumination vector, the scene itself carries out the necessary vector-matrix multiplication.

Using a fixed set of wavelet noise illumination patterns, Peers et al. successively refine a wavelet representing the reflectance field such that the observations can be explained in a least squares sense [Peers and Dutré 05]. This reconstruction problem is similar to compressive sensing where a fixed set of noise patterns are employed to recover a sparse wavelet representation by minimizing the L_1 error [Peers et al. 09]. Sen et al. show that the same set of fixed illumination patterns can be used to recover a sparse representation independent of the chosen basis [Sen and Darabi 09].

In some applications, it is not necessary to recover the complete transport operator. Instead, qualitatively different parts of the light transport are identified. An example is the separation of direct from global light transport effects [Nayar et al. 06]. Providing high-frequency illumination patterns, direct reflections are immediately affected while the global component tends to produce constant response due to smoothing. Similarly, both effects can be separated using time-resolved transient imaging as global effects propagate over longer, more complex paths [Wu et al. 12b]. High-frequency patterns in the angular domain can be employed to separate specular from diffuse reflectance [Lamond et al. 09].

6.6 Measuring Light Transport—Practical Issues

It is good practice to design the measurement process around the scene types to be recorded. One important aspect is invariance: taking many pictures takes significant time, and this may rule out some methods or setups altogether. Human faces, for instance, can be kept motionless for at most a few seconds at a time, and even then, fine structures such as hair may be susceptible to air flow—accordingly, the entire measurement process may not take time in excess of a few seconds, ruling out moving a light source and enforcing a setup with statically configured, quickly switchable light sources [Wenger et al. 05]. But even seemingly inanimate

objects may vary their appearance surprisingly quickly; freshly cut flowers, for instance, may change their shape over a matter of minutes.

Measurement Setup: Illumination

In active approaches, the design of actively controlled illumination plays an important role. Needless to say that external influences are to be minimized, including daylight or other uncontrolled light leakage into the lab in which the recording takes place. Black cloth has been used to great effect [Goesele et al. 00]; it serves a dual purpose of not only isolating the measurement space from the outside, its low albedo also reduces the influence of bounce light, increasing the signal-to-noise ratio. Evenly spread cloth is often preferable to folds: while folds permit less bounce light on average, the bounce light shows visible directional variation, and the folds are revealed in mirror-like reflections, while uniform black cloth may be visually hidden under the camera noise floor. Rough materials are preferable to satin-like appearances as they distribute bounce light more evenly over the outgoing directions.

Both for active and passive approaches, the light source needs to be calibrated: either by recording the incident illumination [Matusik et al. 04] while the primary measurement takes place, or as a separate calibration step. In an active setup, the latter is usually preferable in order to prevent the calibration of the light source from interfering with the measurement itself. For an active illumination measurement pipeline, there is also a choice: one can either observe the light distribution (position and shape of the light source) [Goesele et al. 03] or use technical means to create it (for instance, applying robotics to create a precisely moving light source, or displaying a repeatable pattern on a computer monitor).

Due to the linearity of light transport, it is relatively easy to correct for uneven brightness between illumination sources in software: it is sufficient to multiply a linearized input image to match an exact value. Correcting a deviant position, however, is nearly impossible without knowing scene geometry and reflectance in advance (in which case there is no need to record it anyway). Spectral differences cover a middle ground: for photographic applications, (matrix) color correction may result in satisfactory results; for precise measurements of spectral behavior, the illuminants need to be carefully pre-selected to match precisely.

Depending on the specific illuminants used in recording, variations of light source brightness over the course of the measurement play a role as well: mercury-vapor lamps need a few minutes to heat up, then stay stable for some time, but may subtly fade in brightness over matters of hours. Light-emitting diode (LED) light sources are stable for a longer time and are instantly ready—many white LEDs rely on phosphors, though, to create

a spectrally broad illumination, which may require heat-up times in the order of fractions of a second before the emitted spectrum is stable.

Besides stability, overall brightness of light sources is an important design choice as there is a trade-off on light source brightness and camera exposure time: in principle, less brightness can be compensated for with longer exposure time. The effect of some noise types, however, such as dark current noise (Section 1.3), increases with exposure time. On the other hand, brighter light sources increase the ambient temperature in a usually enclosed space, thus contribute to thermal noise, and may negatively impact the scene to be recorded.

As components for home entertainment systems, digital projectors have recently become relatively inexpensive, and increasingly attractive to use for light transport acquisition purposes. Yet, considered as measurement light sources, they come with challenges of their own. Professional systems still have advantages when it comes to synchronization; at the low end, even seemingly unnecessary problems may be encountered, such as the native output resolution not being available to the input video interfaces. For all technologies, contrast is a serious limitation. Some manufacturers achieve high contrast between successive frames with global shutters. However, for measurement purposes, local contrast plays a bigger role and remains limited: while individual pixels can be reliably switched on, the brightness of a single pixel cannot offset the black level of two million others in a HD projector; while the equivalent of dark frame subtraction—take a picture with the projector instructed to show a black frame—helps in theory, the minute increase in brightness of switching on a single pixel over a uniform, but large frame can rarely be observed with an off-the-shelf digital camera due to its limited in-frame dynamic range (Section 1.3).

Geometrically, projectors can approximately generate light field slices for rays that originate from a single point, the center of projection. As they are designed to illuminate a planar surface with high efficiency, they usually come with large apertures and hence only achieve shallow depth of field, limiting the working volume. Laser projectors, both sweeping line and modulating plane designs, usually do not have this restriction, but come with problems of their own: for one, laser light is coherent and interference effects produce speckle on the camera sensor which may dominate the effects to be measured.

In addition, laser projectors share a problem with RGB LED projectors: the color is mixed from very pure primaries. This makes for excellent color reproduction in projection as perceived by a human observer. However, for measurement purposes, broader spectra deliver better color rendition.

Comparing liquid crystal (LCD) and digital mirror (DLP) technologies reveals differences in how intermediate brightness levels are created. LCDs can maintain constant intermediate levels, but need minimal time to

vary brightness and usually have less local contrast than their DLP counterparts. DLPs, on the other hand, switch practically instantly and have better contrast, but dither between minimal and maximal brightness over space and time. Hence, precise synchronization between projector and camera is an even more important issue. As a minimal measure, the camera exposure time should be chosen as an integer multiple of the projector's frame time. This is especially important for DLP projectors which use a spinning color wheel to create colors, as integer-frame exposure time helps to counter the "rainbow" color artifacts. For multi-projector setups, precise synchronization is important, as well.

Measurement Setup: Supporting Equipment

Beyond illumination devices, other components of the measurement setup need to be picked carefully in order to ensure that light direction and shape are the only variants. This implies high requirements regarding the dependability of the equipment. Tripod mounts must be stable: even if they have a stable "feel," slight movements, which may only amount to a drift of few pixel rows over the course of days, may ruin measurement sequences which take a long time to complete. Cameras without moving parts are preferable to those that need to move a mirror prior to exposure or use mechanical shutters. The measurement lab should ideally be equipped with air conditioning to provide for constant ambient temperature and air humidity.

It should go without saying that in virtually all designs for measuring light transport, a high dynamic range pipeline is required so as to obtain exact reflections (Section 1.5). If highlights on a mirroring surface are clipped, for instance, reflections of dim light situations will come out much too dark.

6.7 Summary

When the task is to resolve the light transport the method of choice depends on the specific global illumination effect that should be resolved. Capturing diffuse or specular reflection of known surfaces is simpler than resolving the material properties contributing to interreflections, subsurface scattering or caustics. Though the light transport can be modeled as a linear operator its complexity in many dimensions still prevent a dense sampling approach—approximating models and reconstruction algorithms are therefore most often employed.

Part II

Reconstruction—Data Processing Techniques

7
Camera Registration from Images and Video

Jan-Michael Frahm and Enrique Dunn

7.1 Introduction

Nowadays, cameras are ubiquitously available and billions of photos and videos are uploaded every year to photo and video sharing sites. Combined with the recent progress in computer vision, this has lead to the development of large-scale 3D modeling from these images with approaches proposed by Snavely et al. [Snavely et al. 06], Agarwal et al. [Agarwal et al. 11], and Frahm et al. [Frahm et al. 10]. Figure 7.1 illustrates a dense 3D model obtained from a photo collection modeling of Frahm et al. [Frahm et al. 10]. These reconstruction methods all rely on the registration of the cameras (determination of their relative/absolute poses at capture and their internal camera parameters) in a common coordinate system. Once the registration is known, the cameras can be leveraged within dense scene geometry estimation (see Chapter 8 for more details on dense depth estimation) to obtain a dense 3D model. Beyond dense depth estimation, camera registration is also required for a broad range of applications like sensor fusion with camera images (see Chapter 9), reconstruction of dynamic structure (see Chapters 11, 12), and more. The automated process of jointly esti-

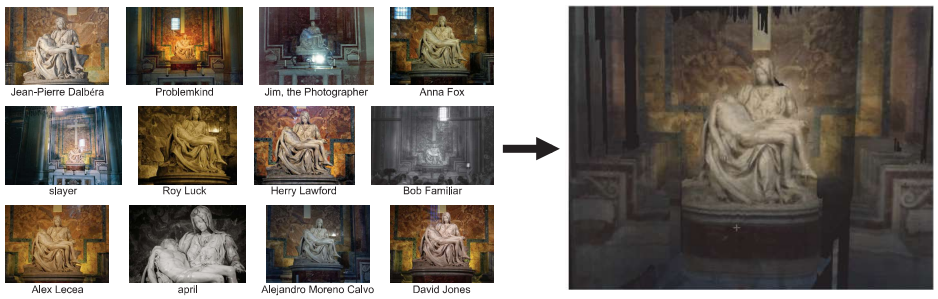


Figure 7.1: An example dense 3D model from an Internet photo collection of Rome, Italy, computed by the method of Frahm et al. [Frahm et al. 10].

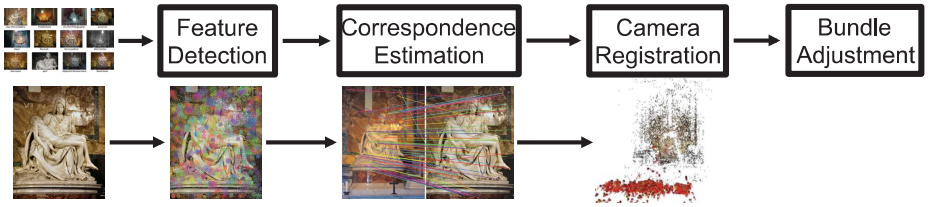


Figure 7.2: Overview of the general processing steps of structure-from-motion methods.

mating sparse scene structure and camera viewing parameters from a set of input imagery, is known as structure from motion. Camera registration is a prevalent task of each part of the structure-from-motion pipeline.

7.2 Structure from Motion Pipeline Overview

Structure from motion algorithms take as input a set of images or video frames and, if available, their associated camera calibration, and determine the relative registration of the cameras during capture. The relative camera registration consists of camera poses of all cameras in a common reconstruction coordinate system, which is related to the world coordinate system by a similarity transformation consisting of a rotation, translation, and scaling of the reconstruction coordinate system. While images and videos are captured in a wide variety of scenes and capture configurations, there are common steps to all structure from motion methods. Most structure from motion techniques perform the following steps (Figure 7.2):

- **Feature detection** determines salient points (features) in the images or video frames. These points are generally expected to be reliably detected across different viewpoints and zoom levels to ensure the detection of the same salient features across different frames. Each feature has an associated descriptor to quantize the characteristics of the feature.
- **Feature matching or tracking** determines the features corresponding to the same 3D point in multiple different images or frames. Matching or tracking determines the correspondence using the descriptors of the detected features in the images and delivers a set of putative correspondences. These putative correspondences typically contain both correct and incorrect correspondences.
- **Robust camera registration** simultaneously determines the correct correspondences from the putative set of correspondences and

the camera poses within the reconstruction coordinate system. Additionally, the 3D points for the correct correspondences are typically triangulated. The set of 3D points is referred to as a sparse point cloud.

- **Bundle adjustment** is an optional non-linear refinement to optimize the camera poses along with the sparse point cloud to obtain more accurate camera poses and sparse 3D features. While this step is optional in small-scale reconstructions, it is recommended for large-scale reconstructions in order to control drift.

Most structure from motion processes follow the above general steps to obtain a camera registration. The differences between various structure from motion systems [Wu 11, Frahm et al. 10, Snavely et al. 06, Agarwal et al. 11, Pollefeys et al. 04] typically result from accounting for different characteristics of captured data (unordered images vs. frames of a video with temporal order) or differences in the desired levels of accuracy or performance. This chapter focuses on the efficient registration of unordered image sets as the most general case of structure from motion.

Local Feature Detection and Description

Estimation of the camera motion between two given images, $\mathbf{I}$ and $\mathbf{I}'$, is at the core of structure from motion methods. While it is easy for humans to perform this task (e.g., if everything in the image moves to the left, the camera moved to the right), for the computer this task is hard as it does not have a global understanding of the image content. Hence, the computer cannot directly infer motion from the captured images $\mathbf{I}$ and $\mathbf{I}'$. Instead, camera motion estimation algorithms rely on the motion of salient 3D points observed in the images to reveal the motion of the capturing cameras. To support this task, structure from motion first detects these salient image features.

The detection of salient features is a long-standing research topic in computer vision and photogrammetry. The desired properties for feature points are:

- **Repeatability:** The same feature can be found in several images despite geometric and photometric transformations.
- **Saliency:** Each feature has to be distinctive and robust to clutter and partial occlusion.
- **Compactness and efficiency:** There are many fewer features than image pixels in each image.
- **Locality:** A feature occupies a relatively small area of the image.

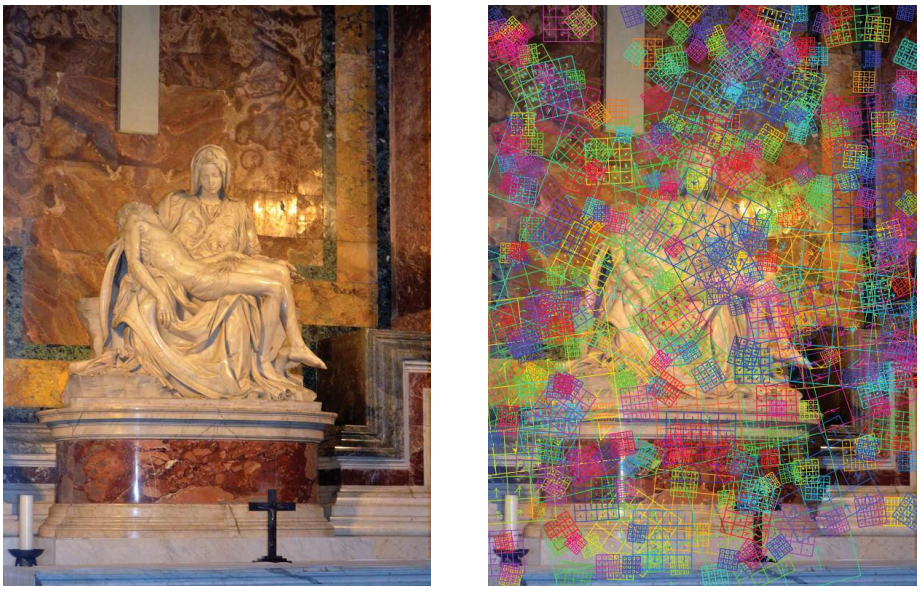


Figure 7.3: Left: Original image of the Pieta in Rome (courtesy David McSpadden). Right: SIFT features detected for the image of the Pieta. The position, orientation and scale of each of the detected features is depicted by an overlayed box.

The above properties may be at odds with each other and will require the design of a feature detection mechanism to consider various performance trade-offs. For example, while achieving repeatability entails invariance/robustness with respect to a given set of image transformations, the presence of such transformations within a single image will hinder the distinctiveness of the detected features given the afforded invariance.

The output of a feature detection process is a sparse set of image pixel positions corresponding to the detected features. In order to perform feature association, a representation describing the (local) appearance properties of each detected feature is required. The set of properties desirable for feature detection is also applicable to feature description mechanisms. There is a wide variety of salient feature detectors/descriptors available in the literature [Lowe 04, Bay et al. 06, Harris and Stephens 88, Matas et al. 04]. One of the most commonly used features is the SIFT feature [Lowe 04], which is robust against in-plane image rotation, changes in the scale of the observed features, and small illumination changes of the scene (Figure 7.3).

While SIFT features are very robust [Mikolajczyk and Schmid 05] their detection is computationally expensive and methods to improve the

computational performance have been proposed. One example of a more computationally efficient feature is the SURF feature [Bay et al. 06], which exercises a similar robustness to rotation, scale, and illumination as the SIFT feature [Heinly et al. 12]. Alternatively, the use of commodity graphics cards has been successfully proposed to perform the feature detection [Sinha et al. 11]. SIFT and SURF features compute a 128-dimensional or 64-dimensional feature descriptor, respectively, for each salient feature point in the image. Given that an image often has several thousand features, the amount of storage required for the feature descriptors is often comparable to the image size. Hence, for large-scale datasets, it is required to reduce the amount of storage to both conserve disk space and to reduce the bandwidth required during matching when the feature descriptors of the images are used to identify potentially corresponding features.

Binary features aim to address the memory efficiency for the descriptors by representing the feature descriptor as a binary vector [Calonder et al. 10, Leutenegger et al. 11, Rublee et al. 11, Alahi et al. 12, Heinly et al. 12]. These binary descriptors typically encode the sign of the local intensity gradient in the image around the 2D feature point for various directions and locations in the vicinity of the feature point. These are then encoded as binary strings to describe the feature. Typical binary descriptors are 128-512 bits and can perform well if chosen appropriately for the application. Heinly et al. [Heinly et al. 12] investigate the performance and the correct choice of binary features for a variety of visual tasks. They show that while no binary feature measures up to the overall performance of SIFT in the large variety of tested scenarios, with careful consideration for the specific needs of any application, binary features can perform similarly well in the appropriate limited scenario.

Once the appropriate features with respect to the targeted application scenario are chosen, these features build the foundation for the next steps. After feature detection and their descriptor computation, the correlation of features across images has to be tackled by the matching or tracking algorithm, which is discussed in more detail in the next section. Popular currently available implementations of the above techniques include [Bradski 00][1] and [Vedaldi and Fulkerson 08].[2]

Feature Matching and Tracking

After detecting the salient feature points in multiple images, the features need to be correlated to obtain the putative matches, which are then used in camera pose estimation. In principle, this correspondence information is obtained through two major classes of techniques. The first class are

[1] http://opencv.org/
[2] http://www.vlfeat.org/

the matching techniques, which detect salient features independently in each image and then correlate their descriptors to obtain the correspondences. The second class are the tracking techniques that detect salient image points in one image and then search for the best corresponding location in the second image. Matching techniques are the more often used for image collections while feature tracking methods are better suited to leverage the temporal correlation observed in video sequences.

Given that for unordered image collections there is typically no prior information about the overlap of any pair of images, no meaningful spatial prior for matching features can be established. Hence, for each feature in the first image, the corresponding feature can be anywhere in the second image, and the correspondence search has to be performed against all features in the image. Given that the feature descriptor is associated with a similarity metric $\mathcal{S}$, the correspondence search obtains the most similar feature point in the second image as the maximum similarity match. For the popular SIFT feature [Lowe 04], with its 128-dimensional feature descriptor, the angle between two feature descriptors $\mathbf{m}_i$ and $\mathbf{m}_j$

$$\mathcal{S}(i,j) = \frac{< \mathbf{m}_i, \mathbf{m}_j >}{\|\mathbf{m}_i\| \|\mathbf{m}_j\|}$$

is used as a similarity metric $\mathcal{S}$. In practice, it is more efficient to normalize the feature descriptors m_i beforehand and use the inner product $< m_i, m_j >$ directly as a similarity metric. For binary feature descriptors the similarity metric is typically the Hamming distance of the feature descriptors [Calonder et al. 10, Leutenegger et al. 11, Rublee et al. 11, Alahi et al. 12].

Using the similarity metric $\mathcal{S}$, the correspondence search then determines the putative match as the feature most similar to the first image's feature (Figure 7.4). Repetitive or duplicate structures in the scene, which have highly similar feature points cause inherent ambiguities in the matching and frequently lead to mismatches (Figure 7.4). To avoid this disturbance, many feature matching methods measure the similarity distance $\mathcal{S}_2$ to the second best match. Lowe [Lowe 04] proposed to remove a match from the pool of putative matches if the ratio $a = \frac{1-\mathcal{S}_1}{1-\mathcal{S}_2}$ of the best similarity $\mathcal{S}_1$ to the second best similarity $\mathcal{S}_2$ measures is too close to one, which means the feature descriptors are very similar and can easily be confused. It is typical to exclude matches whose ratio is higher than 80% [Lowe 04]. Similar ratio tests can be defined for the binary feature's similarity using the Hamming distance [Heinly et al. 12]. Additionally, it is common to test the consistency of the match by matching the feature from the first image to the features in the second image and vice versa. The match is only accepted when the matches from both matchings are consistent. Please note, this doubles the computational effort during matching and is in practice

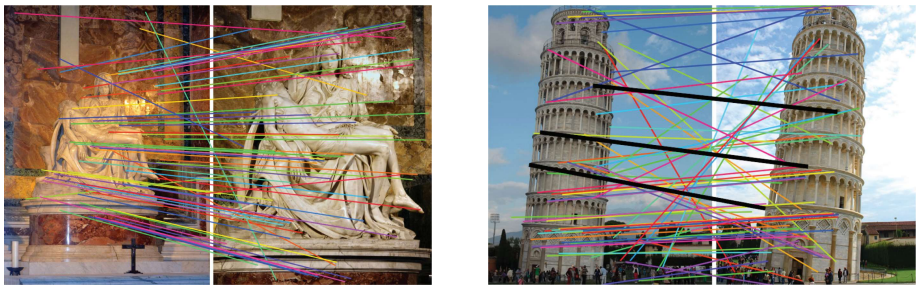


Figure 7.4: Left: Matches for the Pieta in Rome. Right: Disturbing effect of repetitive structures in putative matching (images courtesy of Flickr users David McSpadden, Megan Allen, Luca Semprini, YougoPL).

often not performed to attain a higher processing speed. The remaining putative matches are then used in robust camera estimation (Section 7.2). In uncontrolled image collections it is expected that there are potentially high levels (>70%) of incorrect putative matches. This is due to several contributing factors, such as significant illumination changes, their larger angles of out-of-plane camera motion ($> 30°$), and the occurrence of wider baselines between cameras causing the viewing angle to change significantly.

For videos, there are additional constraints available to further improve the putative matches or reduce the computational effort of the correspondence recovery. In video, the motion between frames is limited and the appearance variation of the scene is typically very limited. Moreover, in practice, the scene illumination can only change slightly between two video frames. Accordingly, in video, feature tracking is exploited for the frame-to-frame correspondence search. In contrast to matching, feature tracking detects the salient feature points in one image, and then actively searches for the corresponding point in the second image, for example using a gradient based search, like it is done in the well-known KLT tracker [Shi and Tomasi 94, Lucas and Kanade 81]. The KLT tracker assumes a small motion (theoretically less than 1 pixel) between images and a constant appearance of the object between two consecutive frames of the video. The stable appearance assumption leads to the widely used brightness constancy assumption

$$\mathbf{I}^{t+1}\left(\mathbf{x}+\frac{\mathbf{dx}}{2}\right)-\mathbf{I}^{t}\left(\mathbf{x}-\frac{\mathbf{dx}}{2}\right)=0, \tag{7.1}$$

where $\mathbf{I}^t$, $\mathbf{I}^{t+1}$ are the frames at time t and $t+1$ and $\mathbf{dx} = (dx, dy)$ is the motion of pixel $\mathbf{x} = (x, y)$ in image $\mathbf{I}^t$ with respect to the image $\mathbf{I}^{t+1}$, i.e., pixel $\mathbf{x}$ in $\mathbf{I}^t$ corresponds to pixel $\mathbf{x}+\mathbf{dx}$ in image $\mathbf{I}^{t+1}$. Leveraging the small motion assumption, Eq.(7.1) can be linearized around the pixel location $\mathbf{x}$. Equation (7.1) considers the brightness of a single pixel, which

is subject to ambiguities due to the similarity of the pixels and due to disturbance by noise. Hence, the KLT tracker assumes a consistent motion of the pixels in a patch $\mathcal{P}$. This leads to the following linear equation system for solving for the unknown motion $\mathbf{dx}$ of the pixel $\mathbf{x}$

$$\sum_{\mathbf{x}\in\mathcal{P}} \begin{pmatrix} \mathbf{I}_x^2 & \mathbf{I}_x\mathbf{I}_y \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y^2 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = 2 \sum_{\mathbf{x}\in\mathcal{P}} \begin{pmatrix} \left(\mathbf{I}^t(\mathbf{x}) - \mathbf{I}^{t+1}(\mathbf{x})\right)\mathbf{I}_x \\ \left(\mathbf{I}^t(\mathbf{x}) - \mathbf{I}^{t+1}(\mathbf{x})\right)\mathbf{I}_y \end{pmatrix}, \quad (7.2)$$

with $\mathbf{I}_x = \frac{\partial \mathbf{I}^{t+1}(\mathbf{x})}{\partial x} + \frac{\partial \mathbf{I}^{t}(\mathbf{x})}{\partial x}$, $\mathbf{I}_y = \frac{\partial \mathbf{I}^{t+1}(\mathbf{x})}{\partial y} + \frac{\partial \mathbf{I}^{t}(\mathbf{x})}{\partial y}$. Eq.(7.2) can be solved to obtain the motion $\mathbf{dx}$ of pixel $\mathbf{x}$. Since this is only valid for small motion in the image, in practice, tracking is performed in a hierarchical way to enable the tracking of larger motions between frames. Kim et al. [Kim et al. 07] proposed an extension of KLT that also accounts for the change in illumination and thus overcomes the strict constraints posed by brightness constancy (Eq.(7.1)).

Due to the explicit correspondence search and the smaller motion, tracking typically produces significantly higher rates of correct correspondences and smaller feature position uncertainties. The uncertainty is, in most cases, reduced to less than a pixel compared to the positional uncertainty of the SIFT based matching, which is typically assumed to be accurate within about four pixels.

After determining the set of putative correspondences between the images or video frames, these correspondences are used to determine the camera pose. The OpenCV library [Bradski 00][3] includes an implementation of the KLT tracker as well as a variety of feature matching frameworks.

Camera Registration

Camera registration within the context of structure from motion entails the estimation of the viewing parameters of a given set of images (Chapter 2). This is achieved through the geometric analysis of their jointly observed scene structure. Intuitively, camera registration strives to determine both the camera pose and internal camera *parameters* that best explain the available image feature *measurements* with respect to a given geometric *model* relating 3D structures and their 2D image observations. The sought internal camera parameters are focal length, principal point, and the skew of the camera (Chapter 1). In general, the level of abstraction suitable for geometric analysis varies according to the image set's cardinality as well as the availability of parameter and scene priors. While pairwise camera analysis based on epipolar geometry can be used to bootstrap projective 3D scene modeling and camera registration, the availability of scene and/or camera

[3] www.opencv.org

intrinsic knowledge can simplify the camera registration problem to one of 3D Euclidean resection [Pollefeys et al. 04, Hartley and Zisserman 03].

These insights are leveraged by structure from motion modules implementing incremental camera pose estimation [Wu 11][4] [Snavely et al. 08b].[5] Specifically, an initial pair or triplet of cameras establishes a common camera coordinate system and the 3D positions of the correct salient feature point matches, enabling subsequent cameras to be registered with respect to the initial reconstruction. There are a few approaches in the literature treating structure from motion as a global problem [Sinha et al. 12]), but for large-scale camera registration this is computationally prohibitive in practice. Camera registration algorithms differ for the cases of uncalibrated and calibrated cameras (i.e., unknown vs. known internal parameters), as in the latter case stronger constraints are available. Camera registration for uncalibrated cameras is more demanding than registration of cameras with known internal camera calibration. Intuitively, the motion of the corresponding features reveals insights on the camera motion between the two images, i.e., if the feature moved to the right the camera moved to the left and similar constraints can be formulated for all degrees of freedom of the camera motion.

Uncalibrated camera registration exploits the fact that an image feature with homogeneous pixel coordinates $\tilde{\mathbf{x}} = (u, v, 1)^\top$, under the pinhole camera model, corresponds to a viewing ray r passing through the camera's optical center. The ray then intersects the image plane at 3D coordinates $\mathbf{K}^{-1}\tilde{\mathbf{x}}$, where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is an upper triangular matrix describing the camera's internal calibration parameters, Eq.(1.10). The camera coordinate system is assumed to be aligned with the first camera, which is at the origin in canonical orientation. To obtain the position of the second camera of the initial pair, structure from motion models the relation between the salient feature points in the first image and its corresponding points in the second image.

The fundamental matrix $\mathbf{F} \in \mathbb{R}^{3\times3}$ linearly maps the viewing ray r of pixel $\tilde{\mathbf{x}}$ in the first camera into a homogeneous 2D line in a second camera, $\mathbf{l}' = \mathbf{F}\tilde{\mathbf{x}}$. The fundamental matrix can be composed as

$$\mathbf{F} = \mathbf{K}'[\mathbf{e}]_\times \mathbf{R}\mathbf{K}^{-1}, \tag{7.3}$$

where the epipole $\mathbf{e}$ is the projection of the second camera's center into the

[4] http://ccwu.me/vsfm/

[5] http://www.cs.cornell.edu/ snavely/bundler/

first camera, $[\cdot]_\times$ represents the matrix formulation of the cross product,[6] $\mathbf{R} \in SO(3)$ is a rotation matrix describing the orientation of the second camera, while $\mathbf{K}$, $\mathbf{K}'$ describe the internal camera parameters for the first and second cameras, respectively. The fundamental matrix has rank two and seven degrees of freedom [Hartley and Zisserman 03]. The seven degrees of freedom result from the fact that a 3×3 matrix has at most nine degrees of freedom. The rank 2 constraint removes one degree of freedom. Additionally, the fact that the fundamental matrix is a projective transformation means that it is scale-invariant, which reduces the degrees of freedom to seven. For a feature point $\tilde{\mathbf{x}}$, the corresponding matching point $\tilde{\mathbf{x}}'$ in the second image is constrained to lie on the line $\mathbf{l}'$, which is the projection of the viewing ray r. This relation is expressed in the epipolar constraint

$$(\tilde{\mathbf{x}}')^\top \underbrace{\mathbf{F}\tilde{\mathbf{x}}}_{\mathbf{l}'} = 0. \tag{7.4}$$

Equation (7.4) provides one constraint for each correspondence $\tilde{\mathbf{x}}$, $\tilde{\mathbf{x}}'$ between a pair of images, enabling the estimation of the fundamental matrix $\mathbf{F}$ from a set of seven or more feature correspondences in general configuration [Hartley and Zisserman 03]. The fundamental matrix can be exclusively computed from feature correspondences in the absence of any priors on the camera parameters. The process of estimating the epipolar geometry between two views is commonly referred to as pairwise geometric verification.

Equation (7.4) is linear with respect to the matrix values $\mathbf{F} = \{F_{jk}\}$ and can be estimated through least squares methods. Moreover, given two corresponding feature positions, $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}'_i$, each epipolar constraint can be expressed in linear form as $\mathbf{a}_i\mathbf{f} = 0$, where $\mathbf{f} = \mathbf{vec}(\mathbf{F})$ is a row-major vectorization of the $\mathbf{F}$ matrix into a column vector, while $\mathbf{a}_i^\top = \mathbf{vec}\left(\tilde{\mathbf{x}}'_i\tilde{\mathbf{x}}_i^\top\right)$ is the vectorization of the outer product of vectors $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}'_i$. Given multiple feature correspondences, the fundamental matrix can be estimated from a linear system of equations of the form

$$\begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_N \end{bmatrix} \mathbf{f} = \mathbf{A}\mathbf{f} = 0. \tag{7.5}$$

Solving Eq.(7.5) should leverage the rank deficiency and scale ambiguity of $\mathbf{F}$. Hartley addressed input normalization in over-constrained estimation,

[6] The cross product matrix for vector $\mathbf{e} = [e_1, e_2, e_3]$ is given by

$$[\mathbf{e}]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}.$$

with eight or more features, to improve numerical stability [Hartley 97]. The fundamental matrix can then be used to obtain the projection matrix of the second camera [Hartley and Zisserman 03]. Note that due to the projective nature of the epipolar geometry formulation, the estimation of the fundamental matrix enables 3D reconstruction only up to a projective transformation of the scene.

Solving Eq.(7.5) assumes strictly error-free feature correspondence estimation, which is rarely attained in practice for scenes captured under uncontrolled settings. Accordingly, robust estimation frameworks like RANSAC [Fischler and Bolles 81] are employed for fundamental matrix estimation.

Robust model estimation is important for obtaining correct reconstructions in the presence of corrupted input measurements. One of the most commonly used robust estimation frameworks is RANSAC, which enables robust parametric model fitting through the joint estimation of model parameters (here the fundamental or essential matrix) and the classification of input data into model compliant (inliers) and non-compliant data (outliers). Algorithm 7.1 provides the pseudo-code for RANSAC.

On a high level the RANSAC algorithm iterates two steps, the *hypothesis generation* and the *hypothesis verification*. In the hypothesis generation RANSAC produces hypotheses for the model leveraging data samples to explore the space of models, i.e., it selects a random sample of the data and computes the model from this data. During the hypothesis verification phase RANSAC uses all data to verify if they support the model and it counts the number of supporting points for each model. After a sufficient number of iterations of hypothesis generation and verification, the best seen model obtained so far is returned.

Hypothesis generation draws sample sets $\mathcal{J}$ of size s from the data $\mathcal{D}$. Then the sample set $\mathcal{J}$ is used to compute a model $M_{\mathcal{J}}$. This model will be the correct model $M_{\mathcal{J}}$ if the sample set $\mathcal{J}$ only contains correct data, called *inliers*, which are compliant with the true underlying model and are only slightly corrupted by noise. If the sample set $\mathcal{J}$ contains erroneous data, called *outliers*, the model $M_{\mathcal{J}}$ will be wrong.

Hypothesis verification aims at determining, which of the generated hypotheses/models $M_{\mathcal{J}}$ is supported by the data. In this context, the support of an individual data point to a model $M_{\mathcal{J}}$ corresponds to a thresholding on the magnitude of the observation residuals with a value Θ (inlier threshold). The points below the threshold Θ form the inlier data points $\mathcal{I}_{\mathcal{J}}$. The set of outliers $\mathcal{O}$ is the complement of the inlier set $\mathcal{I}_{\mathcal{J}}$.

The number of inliers $|\mathcal{I}_{\mathcal{J}}|$ is then used as the criterion for model selection from the generated hypotheses $M_{\mathcal{J}}$ by keeping the best hypothesis

seen so far as the current estimate of the model M. The sequential sampling is terminated once the current number of iterations guarantees that, with probability ρ, a good model has been sampled. The number of required samples h is defined by:

$$h = \frac{\log(1-\rho)}{\log(1-\epsilon^s)}, \tag{7.6}$$

with $\epsilon = \frac{|\mathcal{D}|}{|\mathcal{I}|}$ being the fraction of inlier data points in the total data $\mathcal{D}$. Hence, the required number of iterations h is a function of the fraction of inliers in the input data, the size of the sample set $\mathcal{J}$ and the desired level of confidence. Moreover, lower inlier ratios ϵ will reduce the probability of each sample to be a correct sample of data points not corrupted by noise. Similarly, a large data sampling set $\mathcal{J}$ will reduce the probability of finding an ensemble of data points exclusively comprised by inliers. Accordingly, the use of minimal sampling sets is required for efficiency-driven applications. For example, the fundamental matrix estimation based on the minimal sample of $n = 7$ points (sampling subset) is generally preferred over the more numerically stable 8-point method in the context of RANSAC-based robust estimation.

As explained above, by leveraging the fundamental matrix a projective camera registration can be established, which can then be upgraded into a Euclidean reconstruction. If the camera's internal calibration parameters are known, a Euclidean reconstruction can be established directly.

Data: dataset $\mathcal{D}$ for model fitting, confidence level ρ
Result: $M, \mathcal{I}$
$j = 0$, $h = \infty$, $\mathcal{I} = \emptyset$
repeat
  Hypotheses generation
  - chose random minimal sample set $\mathcal{J}$ of size s from data $\mathcal{D}$
  - compute model $M_{\mathcal{J}}$ from minimal sample $\mathcal{J}$
  Hypothesis evaluation
  - determine inlier set $\mathcal{I}_{\mathcal{J}}$
  if $|\mathcal{I}_{\mathcal{J}}| > |\mathcal{I}|$ **then**
    $M = M_{\mathcal{J}}$, $\mathcal{I} = \mathcal{I}_{\mathcal{J}}$
    $\epsilon = \frac{|\mathcal{D}|}{|\mathcal{I}|}$, $h = \frac{\log(1-\rho)}{\log(1-\epsilon^s)}$
  end
  $j+ = 1$
until $h < j$;

Algorithm 7.1: RANSAC algorithm

Calibrated camera registration is based on the more constrained essential matrix $\mathbf{E}$ instead of the fundamental matrix. The essential matrix $\mathbf{E}$ describes the relationship between a pair of views in a Euclidean setting, instead of the projective context provided by a fundamental matrix $\mathbf{F}$. The essential matrix is given by

$$\mathbf{E} = \mathbf{K}'^{\top}\mathbf{F}\mathbf{K} = [\mathbf{t}]_{\times}\mathbf{R}, \tag{7.7}$$

where $\mathbf{t}$ describes a translation vector between cameras. Moreover, $\mathbf{E}$ has only five degrees of freedom and can be estimated from five correspondences [Nistér 04], which leads to a more efficient RANSAC-based estimation. Given that the essential matrix avoids the degeneracy of the fundamental matrix for 3D points that are on a plane, it is always advised to employ essential matrix estimation for calibrated cameras. Even if there is no accurate calibration available, many digital cameras nowadays provide an estimate of their internal calibration through the EXIF data embedded into the image data file. In practice, approximating the principal point to be located at the image pixel center and stipulating a focal length within 50% of the ground truth nominal value is generally sufficient for attaining estimates of the essential matrix [Nistér 04]. Alternatively, the work of Bougnoux [Bougnoux 98], Sturm [Sturm 01], and Hartley [Hartley 93] has explored the estimation of focal lengths strictly from analysis of the fundamental matrix, encompassing a diverse set of scenarios.

Given an essential matrix estimate, the pairwise relative camera poses can be estimated by a matrix decomposition combined with oriented geometry [Hartley and Zisserman 03]. In this way, a registration of both cameras in the same coordinate system is attained. One camera defines the origin of the coordinate system in canonical orientation, while the displacement vector and the relative orientation of the other camera are denoted by $\mathbf{t}$ and $\mathbf{R}$ in Eq.(7.7). Please note that the resulting coordinate system is of arbitrary scale with respect to the world coordinate system, yielding an ambiguity in the magnitude of the baseline vector $\mathbf{t}$.

Two-view triangulation is the process of inferring the 3D structure of the image features, given the estimated relative camera motion described by the essential matrix. Given a pair of image measurements $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ in each image, and the pairwise essential matrix $\mathbf{E}$, the geometry of both viewing rays r and r' in a common 3D reference frame can be determined, as well as their 3D intersection. In practice, image feature positions are corrupted by measurement errors, and thus rays do not intersect in 3D. This is why 3D triangulation is usually phrased as residual minimization. Hartley and Sturm [Hartley and Sturm 97] proposed a closed-form formulation to determine the optimal image corrections compliant with a specified

epipolar geometry. Alternatively, Kanatani et al. [Kanatani et al. 08] and more recently Lindstrom [Lindstrom 10], have proposed iterative solutions based on non-linear optimization. In their work, feature measurements are used as initialization priors. The cost function being optimized is the sum of squared residuals, and the search space is (implicitly) defined over candidate 3D positions subject to the constraints defined by the essential matrix.

Incremental 3D reconstruction systematically augments an initial sparse 3D model obtained from a camera pair using the fundamental or the essential matrix. The sparse 3D reconstruction employs the feature correspondences and intersects their associated viewing rays to compute the position of sparse features in 3D. The 3D reconstruction process has so far 1) defined a Euclidean 3D reference coordinate system (up to scale), 2) identified a set of geometrically consistent 3D landmarks (i.e., triangulated features), and 3) determined the spatial relationships between the input images (i.e., epipolar geometry). The augmentation of our existing reconstruction can now leverage the estimated Euclidean structure and perform pairwise camera calibration by solving the perspective three point problem (P3P) [Haralick et al. 94]. More specifically, given a set of three 3D landmarks and their projections (i.e., the corresponding 2D feature points) to a calibrated camera with unknown pose, it is possible to use the known 3D positions and the angle among the corresponding viewing rays in the new camera to solve for the distance along each viewing ray and estimate the rigid motion transformation of the camera. The typical framework is to sequentially execute different RANSAC instances attempting to register a new input image against each of the existing registered cameras.

Bundle Adjustment

In the following it is assumed that the data used for camera registration, comprising input feature correspondence observations, estimated output camera registrations, and sparse 3D points are denoted by $\mathcal{O}$, $\mathcal{C}$, and $\mathcal{S}$, respectively. Initial estimates for an individual camera $\mathbf{c}_i \in \mathcal{C}$ and a 3D scene point $\mathbf{s}_j \in \mathcal{S}$ are typically obtained through pairwise camera registration techniques. The incremental nature of these registrations can lead to global estimation inconsistencies. The process by which sets of camera estimates $\mathcal{C}' \subset \mathcal{C}$ and structure estimates $\mathcal{S}' \subset \mathcal{S}$ are jointly refined through non-linear optimization techniques is known as bundle adjustment. In practice, bundle adjustment can alternatively serve as 1) a post-processing step with the goal of refining a given camera registration, and/or 2) a systematic geometric consistency enforcement module during incremental camera registration. More detailed discussions within the context of photogrammetric measurements can be found in [McGlone 13], while the survey presented

by Triggs et al. [Triggs et al. 00] provides a discussion from the perspective of the computer vision community.

An observation residual is defined as $\mathbf{r}_{ij} = \mathbf{o}_{ij} - f(\mathbf{c}_i, \mathbf{s}_j)$, where $f(\cdot)$ describes the image formation model evaluated for a given camera and structure parameter instance, while the camera and 3D feature indices are denoted by i and j, respectively. To minimize the image reprojection errors a weighted least squares optimization problem is defined as follows:

$$\min_{\mathbf{c},\mathbf{s}} \sum_{\forall\{i\times j\}:\exists\mathbf{o}_{ij}} \mathbf{r}_{ij}^{\top}\mathbf{W}_{\mathbf{o}_{ij}}\mathbf{r}_{ij}, \tag{7.8}$$

where $\mathbf{W}_{\mathbf{o}_{ij}}^{-1}$ approximates the measurement covariance matrix. Equation (7.8) can be iteratively solved using an approximated second-order Gauss–Newton refinement step $\Delta_{\mathbf{c},\mathbf{s}}$ defined by

$$\left(\mathbf{J}^{\top}\mathbf{W}_{\mathbf{o}}\mathbf{J}\right)\Delta_{\mathbf{c},\mathbf{s}} = -\mathbf{J}^{\top}\mathbf{W}_{\mathbf{o}}\,\mathbf{r}, \tag{7.9}$$

where $\mathbf{J} = \left[\frac{\partial f(\mathbf{c}_i,\mathbf{s}_j)}{\partial \mathbf{c}} \;\; \frac{\partial f(\mathbf{c}_i,\mathbf{s}_j)}{\partial \mathbf{s}}\right]$ denotes the Jacobian of the image formation model with respect to the camera and structure parameters. The above formulation entails the inversion of the normal equations defined by $\mathbf{J}^{\top}\mathbf{W}_{\mathbf{o}}\mathbf{J}$, for which efficient solvers leveraging the equation's block (and possibly sparse) structure can be utilized [Agarwal et al. 10, Wu et al. 11]. Nevertheless, performing such global refinement in an incremental setting may become a limiting computational overhead for large 3D reconstructions. In this respect, the concept of *windowed* (i.e., reduced local neighborhood) camera subset selection (i.e., $\mathbf{c} \subset \mathbf{C}$, where $|\mathbf{c}| \ll |\mathbf{C}|$) enables a trade-off among estimation reliability and computational efficiency. Accordingly, temporal windowing is generally applied to video-based reconstructions, while spatial windowing is generally applied to unordered input image sets.

After introducing the general building blocks of structure from motion, the next section will discuss some of the modifications necessary for large-scale structure from motion based on these building blocks.

7.3 Scalable Structure from Motion

This section discusses some of the challenges and considerations when doing structure from motion of large-scale image sets. In this context, the main (often competing) objectives are scalability and robustness. While the former entails the ability to operate on Internet-scale input image sets (currently comprising thousands to millions of images), the latter implies the ability to operate on heterogeneously captured input data. In practice, both of these objectives are tightly coupled and hinge on diverse design and

implementation considerations. Moreover, the relative maturity and effectiveness of state-of-the-art structure from motion technologies shifts the focus of large-scale implementation to developing effective data association and process management modules. Given that the use of an incremental structure from motion framework (as described in Section 7.2) is the *de facto* design choice for large-scale image sets, the remainder of this section discusses the different data association considerations within this framework.

Before discussing scalability, two major considerations are discussed that improve the robustness of structure from motion: the selection of the first pair of images from which to initialize, and the selection of the next best view to add to the reconstruction.

initial pair selection is critical for a robust and well-behaved structure from motion process. The initial pair needs to have a sufficient baseline to ensure that any triangulated 3D point has sufficient accuracy. On the other hand, if the baseline is too large, the triangulation produces very accurate 3D points but the matching ability degrades due to the appearance change. Moreover, the optimal initial pair would produce a high and well-distributed number of 3D points that match with a large number of other images in the image collection. In practice, the initial pair has to find a compromise between these competing goals of matching, triangulation, and scene structure. Beder and Steffen [Beder and Steffen 06] consider the uncertainty of the triangulated points for selecting the initial pair, which leads to stable structure from motion. Hence, their method focuses on increasing the baseline of the initial pair. In contrast, Snavely et al. [Snavely et al. 06] propose to initialize structure from motion by choosing the pair with the maximal number of corresponding points whose correspondences are not explained by a planar scene or a camera rotation. Hence, they emphasize the importance of the scene structure over the accuracy of the triangulated points, which is improved through their subsequent processing steps. A combination of these criteria is proposed by Raguram et al. [Raguram et al. 11]. They optimize the number of inliers of a pair and the uncertainty of the triangulated points simultaneously. After selecting the initial pair, structure from motion needs to decide in which order to register the remaining views, which is done by the next best view selection.

Next best view selection for incrementally growing the structure from motion is important to achieve a stable reconstruction. For example, selecting a weakly connected view could lead to an erroneous registration for the next best view. This corrupted reconstruction can then influence the succeeding registrations through perturbed 3D points used in the registration.

Similar to the selection of the initial pair, the next best view should have a sufficiently high number of 3D points that correspond to the 2D salient feature points visible in the camera view. Additionally, the uncertainties of those 3D points should be as small as possible to allow accurate registration of the view. For example, in Raguram et al. [Raguram et al. 11] and Snavely et al. [Snavely et al. 06] the next best view is chosen as the view with the highest number of visible 3D points. The 2D-3D point correspondences are obtained by using the pairwise viewing registrations and their 2D-2D matches to predict the visibility of the 3D points (Section 7.2). This criterion favors high scene overlap over enforcing higher accuracy of the 3D points.

Scalability is critical for large-scale structure from motion in order to reduce the computational complexity. Please note that most of the above selection strategies consider the availability of pairwise registrations of all images. These pairwise registrations are, for example, described by the essential matrix of the pair and its feature inliers. Computing these relations and inliers for all possible pairs of images in a collection is computationally prohibitive, even for collections of a few thousand images. One such example is the seminal work of Snavely et al. [Snavely et al. 06], where the registration of approximately three thousand images required about two weeks of computation time. It can be observed that in larger photo collections, where a large number of pairs does not overlap at all [Frahm et al. 10] or are not required for a stable reconstruction [Snavely et al. 08c]. Hence, Agarwal et al. [Agarwal et al. 11] propose to use a vocabulary tree search [Nistér and Stewenius 06] for identifying overlapping images and combine the search with query expansion [Chum et al. 07] to increase the number of overlapping images returned by the search. Frahm et al. [Frahm et al. 10] propose the iconic scene graph to limit the search for overlapping pairs. They employ iconic images, i.e., representative images for subsets of the database, to represent the viewpoints in the scene. Both Agarwal et al. [Agarwal et al. 11] and Frahm et al. [Frahm et al. 10] improve the scalability with the latter technique, reaching overall linear complexity for image registration, which enables the scaling to the processing of millions of images on a single day using a single PC.

7.4 Summary

This chapter discussed the underlying principles of structure from motion, which is an important enabling technique for image-based reconstruction-from photo and video collections. These principles are common to the

large body of state-of-the-art algorithms. The discussed methods span from small-scale structure from motion to methods for large-scale structure from motion for crowd sourced data. The latter have to overcome more significant challenges with respect to robustness and scalability to meet the demand of reconstructing from thousands [Snavely et al. 06] to millions of images [Frahm et al. 10]. Yet, the discussed methods only represent a small fraction of the state-of-the-art structure from motion methods and do not represent the full range of methods [Pollefeys et al. 04, Hartley and Zisserman 03, Beardsley et al. 97, Lhuillier and Quan 05, Dellaert et al. 00, Wilson and Snavely 13].

8
Reconstruction of Dense Correspondences

Martin Eisemann, Jan-Michael Frahm,
Yannick Remion, and Muhannad Ismaël

8.1 Introduction

This chapter concentrates on dense image correspondence estimation with a special focus on stereo. Images are the basic input for a vast majority of algorithms dealing with the reconstruction of the real world. To analyze a scene from a collection of images it becomes inevitable to put these images into correspondence. These correspondences then form the basis for many subsequent analyses, including camera calibration, stereo and 3D reconstruction, motion information, scene flow, and others. While some of these tasks like camera calibration require only sparse correspondences between the images (Chapter 7), others require per-pixel correspondence, also known as dense correspondence estimation.

Humans are extremely good at solving the correspondence problem which most of them do all the time during depth perception. Basically, the eyes serve as two cameras, slightly displaced, with respect to each other, that capture the surroundings from two different viewpoints. When focusing on an object at a certain distance one has already computed an estimate of the distance in the brain and therefore of the object's position in space. It turns out the same problem is quite difficult for a computer and has been researched for several decades now.

The difficulty in correspondence estimation is caused by several factors: images are often corrupted by sensor noise, e.g., when recorded in a poorly lit environment (Section 1.1); the captured scene signal is discretized and represented by some finite image resolution; not every pixel actually has a corresponding partner in the other views as it might be occluded; and ambiguities due to the absence of texture are difficult to solve.

If one can solve the dense correspondence problem a variety of different applications becomes possible especially in the field of computer vision. Robot navigation and autonomous cars require depth perception to avoid

obstacles [Giachetti et al. 98, Kastrinaki et al. 03]. Quality assurance in industrial applications is often based on stereo algorithms to detect cracks and ridges in manufactured products. Reconstruction of urban environments from images has recently gained a lot of interest in the research community [Gallup et al. 07, Frahm et al. 10]. The dense correspondences allow for video editing [Adobe Systems Inc. 13, The Foundry 13], super-resolution [Irani and Peleg 91], video stabilization [Matsushita et al. 06], to interpolate between images [Chen 95, Lipski et al. 10a], e.g., to create bullet time effects made famous in the blockbuster movie *The Matrix* and for specific tracking applications in graphics, e.g., the local pose optimization for texture correspondence matching in Chapter 11 is related. Disparity remapping based on the correspondences and reconstructed depth becomes important to avoid visual fatigue in stereoscopic cinema [Devernay and Beardsley 10].

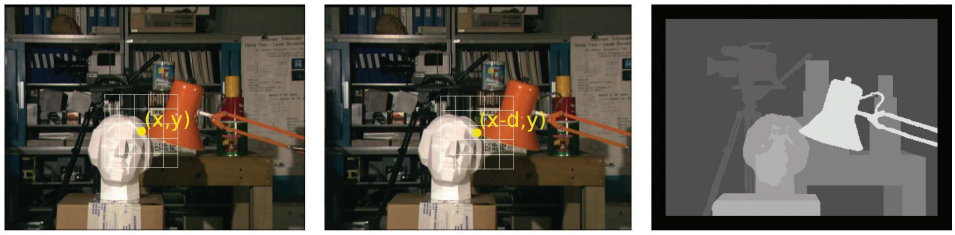
The following will give a hands-on guide on how to compute dense correspondences between images. After a short overview of current state-of-the-art approaches (Section 8.2), a robust solution to the correspondence problem is described and extended (Section 8.3). It is described how to compute correspondences from multiple images (Section 8.4), and means to speed up the computations using graphics hardware are presented (Section 8.5).

8.2 Overview

This section gives a brief overview of different approaches dealing with the dense correspondence problem. The goal is to find the best corresponding (sub-)pixel position in neighboring views for every pixel of a reference image, if such corresponding positions exist.

The algorithms dealing with the correspondence problem can be broadly classified into two categories: stereo and optical flow. Intrinsically the problem is the same for both of them, finding good correspondences between the views, but they differ in the premises. Stereo can be seen as a special case of optical flow, where correspondences are searched along the same scanline (or epipolar line), reducing the solution space from 2D to 1D. The following will give a short overview of the most seminal papers in both categories and their contributions.

Stereo In analogy to the human eyes, the input to classic binocular stereo algorithms are two images $\mathbf{I}^l$ and $\mathbf{I}^r$, a left and a right one. The task is to find for every pixel $\mathbf{p}$ with pixel coordinates (x, y) in the left image a corresponding pixel $\mathbf{q}$ in the right view with pixel coordinates $(x - d_{\mathbf{p}}, y)$.



(a) (b) (c)

Figure 8.1: Dense correspondence estimation in stereo for the Middlebury Tsukuba dataset [Scharstein and Szeliski 02]. (a) The task is to find for each pixel at any position (x, y) in the left view (b) a corresponding position $(x - d, y)$ in the right view and encode the result in (c) a disparity map from which 3D coordinates can be reconstructed. In the stereo setting, the corresponding pixels lie on the same scanline, whereas in the more general problem of optical flow estimation the correspondence can be any position within the right view. Instead of comparing single pixel values, comparing neighborhoods of pixels (shown as the overlaid grid) results in higher robustness.

$d_{\mathbf{p}}$ is called the disparity of pixel $\mathbf{p}$. The disparity information is typically saved in an intensity image, the so-called disparity map $\mathbf{D}$, where low/dark values encode low disparity and high/bright values encode high disparity (Figure 8.1(c)).

In stereo one generally distinguishes between local and global methods. In the first category local areas of one image are matched to local areas in the corresponding view, often called support regions. The difficulty lies in the choice of the support region as matching single pixels is highly ambiguous in most scenes. Simple rectangular windows around the pixel under consideration can be efficiently implemented [Hirschmüller et al. 02, Mühlmann et al. 02] but it can be difficult to choose the right size. By shifting the center position of the window and testing different sizes [Fusiello et al. 97] or by deactivating parts of the support region [Hirschmüller et al. 02, Veksler 02] one can hope that at least one constellation does not overlap with a depth discontinuity. This otherwise poses a matching problem as in many cases a depth discontinuity marks the separation line between two objects with different disparities and, therefore, a different amount of motion in image space from the left to the right view. The research community has thus investigated methods to find a good support region, with different criteria on how much influence each pixel inside this region should have on the final result [Hosni et al. 13].

One key component, and a breakthrough for local methods in recent years, has been the introduction of adaptive support weights [Yoon and

Kweon 05]. The idea is to adjust the influence of neighboring pixels on the final matching cost based on a similarity metric, most often color and spatial similarity. [Yoon and Kweon 05]'s bilateral weighting scheme is based on a Gaussian distribution depending on the spatial proximity and proximity of intensity values. To overcome the problem of spatially close but distinct objects influencing each other, the spatial proximity can be exchanged with a geodesic distance [Hosni et al. 09].

Unfortunately, the computation of adaptive support weights is costly if implemented in a naive way. To speed up the aggregation step it can be converted to an image filtering procedure. It turned out that the bilateral weighting scheme of [Yoon and Kweon 05] is equivalent to applying a cross-bilateral filter or derivations of it to the x, y-slices of a cost volume [Hosni et al. 11b, Richardt et al. 10, Zhang et al. 10a, Ju and Kang 09]. To further speed up the computation, the pixel-wise matching for fixed disparities can be elegantly formulated as a plane-sweeping algorithm on the GPU [Yang and Pollefeys 03, Gallup et al. 07, Zach et al. 08] allowing for real-time stereo implementations.

An implicit assumption made by the aforementioned techniques is that each local support region is basically a patch with fronto-parallel orientation to the image plane of the reference view. Treating the slices in the cost volume not as virtual planes representing a certain disparity but as real 3D planes in the scene one can easily use rotated versions of these slices to compute the matching cost for slanted surfaces [Gallup et al. 07]. The computation times, however, increase linearly with the number of orientations used. Therefore, [Zhang et al. 08] propose to iteratively refine the disparities and orientations in a feedback loop. Another alternative is to initialize each pixel with a random orientation and disparity and propagate good matches to neighboring pixels based on a PatchMatch update scheme [Bleyer et al. 11a].

The second category of stereo algorithms forms the so-called global methods. Global stereo methods pose the matching problem as an energy minimization problem which is usually of the following form:

$$E(\mathbf{D}) = E_{\text{data}}(\mathbf{D}) + \alpha \cdot E_{\text{smooth}}(\mathbf{D}) \quad , \tag{8.1}$$

where $\mathbf{D}$ is the current estimate of the disparity map. The goal is to find $\mathbf{D}$ that produces the lowest energy. E_{data} in this context is a photo-consistency measure that can be equal to the matching function of the local methods but is traditionally simpler. Instead of implicitly stating a smoothness function in the form of a support region, as in the local approaches, here the smoothness is explicitly expressed within the error formulation as E_{smooth}. This regularization of the solution can be especially useful for textureless regions as it basically smoothes out the solution.

Several optimization approaches have been proposed to minimize Eq.(8.1) through dynamic programming [Veksler 05, Bleyer and Gelautz 08], graph-cuts [Boykov et al. 01, Hong and Chen 04, Bleyer and Gelautz 07] or belief propagation [Sun et al. 03, Yang et al. 06b, Taguchi et al. 08].

Interestingly, the usage of tree-reweighted message passing (TRW) and a comparison to ground truth results revealed that modern optimization algorithms yield energies that are actually lower than that of the ground truth solution [Szeliski et al. 08]. This indicates that the model in Eq.(8.1) is actually a limiting factor. Further advances, therefore, need to extend the model. Explicit occlusion handling or enforcing symmetrical matches between the input images was used, e.g., in [Kolmogorov and Zabih 01, Lin and Tomasi 04, Sun et al. 05, Woodford et al. 09]. Truncating the smoothness term to a user-defined maximum value favors large jumps in the disparity map instead of many small changes [Hirschmüller 05, Sun et al. 05, Yang et al. 06a]. Segmentation-based methods presegment the image into patches of coherent color and match whole segments at once [Deng et al. 05, Hong and Chen 04, Zitnick et al. 04]. The idea is that in many cases depth discontinuities coincide with segment borders. An extension of segmentation-based stereo is object-based stereo which matches semantic objects instead of single colored patches. In this way it becomes possible to handle even semi-occluded surfaces [Bleyer et al. 11b]. Extending the idea of object-based stereo one can estimate simple 3D approximations for the different objects [Bleyer et al. 12]. On the basis of these higher semantic concepts one can add sophisticated additional constraints to the optimization, for instance to prevent intersections between the objects or to add a gravity constraint.

Optical flow The problem of optical flow estimation is strongly related to the stereo problem and several of the aforementioned algorithms are applicable to both. Basically, optical flow estimation is a generalization of the stereo problem from a 1D solution space, the disparity map, to a 2D solution space, the flow or motion field. While stereo algorithms aim at reconstructing correspondences between images captured at the same instance in time, optical flow allows to track the motion of pixels also across the time dimension, e.g., in a video.

During the last 30 years, hundreds of research papers have been published in the field of optical flow and various surveys and benchmarks cover and compare the state-of-the-art [Barron et al. 94, Baker et al. 11]. The seminal work of [Horn and Schunck 81] and [Lucas and Kanade 81] laid the foundations for the algorithms to follow. Interestingly, similar to stereo, one can distinguish global and local approaches to the optical flow problem, explicitly enforcing smoothness in the solution [Horn and Schunck 81]

and assuming local constancy within a window around each pixel [Lucas and Kanade 81]. Neither assumption of smoothness holds at motion boundaries for which robust [Black and Anandan 96, Zach et al. 07] and anisotropic regularizers [Nagel and Enkelmann 86, Werlberger et al. 09, Sun et al. 10, Zimmer et al. 11] have therefore been proposed. To reduce the influence of outlier pixels caused by brightness changes and sensor noise the simple data terms based on color-constancy assumption are mostly replaced by robust penalizer functions [Black and Anandan 96, Brox et al. 04, Zach et al. 07] or pixel-descriptors [Mileva et al. 07, Liu et al. 08].

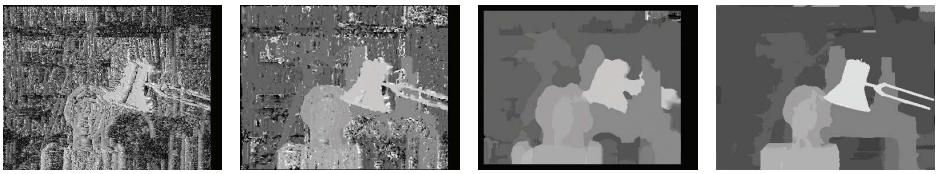
To cope with fast motion, scale-space approaches [Anandan 89] and iterative warping schemes [Alvarez et al. 00, Brox et al. 04] make use of image pyramids to find corresponding pixels. As downsampling only works well for sufficiently large objects several search schemes have been proposed in the literature that either perform a full search [Steinbrücker et al. 09, Linz et al. 10a, Lipski et al. 10b, Hosni et al. 11b] or use tracked features as reliable priors for the optimization [Brox and Malik 11]. In a more hardware-based approach [Lim et al. 05] make use of a high-speed camera to reduce the per pixel displacement to less than a pixel.

Probably due to its success in stereo, explicit occlusion handling has been introduced to optical flow estimation as well. The occlusion detection thereby is either based on the optimization residual and divergence of the flow [Xiao et al. 06, Sand and Teller 06], the symmetry of forward and backward flow [Alvarez et al. 07, Linz et al. 10a, Lipski et al. 10b], or is integrated in the image formation model making use of alternate exposure images [Sellent et al. 11] by alternate capturing of long- and short-exposed images in a video.

8.3 Dense Correspondence Estimation

In the following, an approach is described to compute dense correspondences between two images. The algorithm is mainly based on the fast cost-volume filtering by [Hosni et al. 11b] which is one of the top ranked local methods for stereo and which is also applicable to the more generalized optical flow problem.[1] For simplicity it is assumed that the images have already been rectified, i.e. corresponding points lie on the same scanline (Figure 8.1). Otherwise, it is assumed that standard rectification algorithms are applied first [Hartley and Zisserman 03].[2] These are generally based on the camera registration procedures described in Chapter 7. These constraints will be loosened in the later part of this chapter (Section 8.5).

[1] Code is available at https://www.ims.tuwien.ac.at/publications/tuw-210567
[2] Code is available at http://www.robots.ox.ac.uk:5000/~vgg/hzbook/code/.



(a) (b) (c) (d)

Figure 8.2: Different dissimilarity functions. (a) Pixel-wise matching solely based on color/intensity is highly ambiguous. (b) A 3×3 correlation window is still noisy. (c) A 21×21 correlation window results in edge fattening. (d) The cost filter method of [Hosni et al. 11b].

The basic task of estimating a disparity map $\mathbf{D}$ can be formulated for each pixel $\mathbf{p}$ as

$$d_{\mathbf{p}} = \underset{0 \leq d \leq d_{\max}}{\operatorname{argmin}} \, c(\mathbf{p}, \mathbf{p} - d) \ . \tag{8.2}$$

The term $d_{\max}$ is a user-defined constant which must be larger than the expected maximum disparity. Note that due to rectification d is always a positive value or 0. In case the disparity map for the right image is to be computed $\mathbf{p} - d$ in Eq. (8.2) is replaced by $\mathbf{p} + d$. For simplicity, only disparity computations for the left image are considered. The simplified notation $\mathbf{p} - d$ denotes the pixel 2D pixel position $(x_p - d_p, y_p)$ where (x_p, y_p) are the pixel coordinates of pixel $\mathbf{p}$. The symbol c denotes a cost / dissimilarity function.

Dissimilarity functions To find an appropriate disparity $d_{\mathbf{p}}$ for each pixel $\mathbf{p}$ one needs to find a suitable dissimilarity function c in Eq.(8.2). The probably most simple one would be a naive per-pixel matching, that is, $c(\mathbf{p}, \mathbf{q}) = |\mathbf{I}^l_{\mathbf{p}} - \mathbf{I}^r_{\mathbf{q}}|_2$ where $\mathbf{I}^l_{\mathbf{p}}$ denotes the pixel intensity of $\mathbf{I}^l$ at pixel position $\mathbf{p}$ and $|\cdot|_2$ is the Euclidean distance between the two vectors. But matching only simple intensity values is highly ambiguous and leads to very noisy results (Figure 8.2(a)).

Instead of matching single pixels one can match small image patches centered at $\mathbf{p}$. In this case the cost function becomes

$$d_{\mathbf{p}} = \underset{0 \leq d \leq d_{\max}}{\operatorname{argmin}} \sum_{\mathbf{q} \in \mathbf{W}_{\mathbf{p}}} c(\mathbf{q}, \mathbf{q} - d) \ \ , \tag{8.3}$$

where $\mathbf{W}_{\mathbf{p}}$ is a square window centered at $\mathbf{p}$, and c as defined above. Figures 8.2(b) and 8.2(c) show the resulting disparity maps using correlation windows of the size 3×3 and 21×21 pixels, respectively. The choice of a right size has a notable influence on algorithmic performance, and no single

window size generally works for all cases. While smaller window sizes capture finer details, matching scores can be highly ambiguous. Larger window sizes, on the other hand, lead to edge fattening around discontinuities and oversmooth results. What is needed is an adaptive support weight that adjusts the shape of the window or, in other words, reduces the influence of pixels that do not belong to the same object as pixel $\mathbf{p}$.

Adaptive support weights Adaptive support weights adjust the influence of each individual pixel considered in the matching process. This can be formulated as a simple extension to Eq.(8.3)

$$d_{\mathbf{p}} = \underset{0 \leq d \leq d_{\max}}{\operatorname{argmin}} \sum_{\mathbf{q} \in \mathbf{W_p}} w(\mathbf{p}, \mathbf{q}) \cdot c(\mathbf{q}, \mathbf{q} - d) \quad , \tag{8.4}$$

where $w(\mathbf{p}, \mathbf{q})$ is a weighting function which should return a value of 1 if $\mathbf{q}$ has the same disparity as $\mathbf{p}$, and 0 otherwise. As this disparity is not known, the weight is usually based on some heuristic that represents the probability that pixel $\mathbf{q}$ exhibits the same disparity as $\mathbf{p}$. The most common assumption made is that pixels close to $\mathbf{p}$ are more likely to belong to the same object and have a more similar disparity than pixels farther away. Additionally, pixels with similar color are more likely to belong to the same object than those with dissimilar color values. The bilateral weighting scheme proposed in [Yoon and Kweon 05] expresses this correlation as

$$w_b(\mathbf{p}, \mathbf{q}) = exp\left(-\left(\frac{c_c(\mathbf{p}, \mathbf{q})}{\sigma_c} + \frac{c_s(\mathbf{p}, \mathbf{q})}{\sigma_s}\right)\right) \quad . \tag{8.5}$$

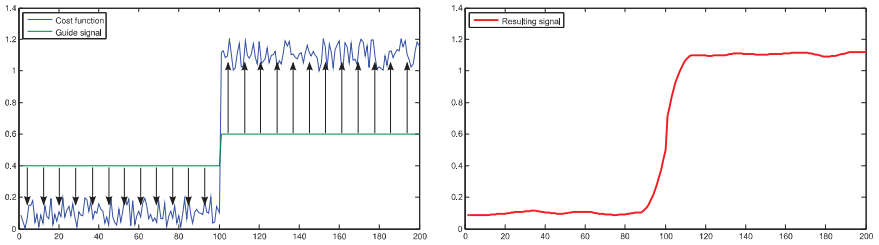
The function $c_c(\mathbf{p}, \mathbf{q})$ denotes the similarity in color defined as the Euclidean distance of pixels at position $\mathbf{p}$ and $\mathbf{q}$ in RGB space, whereas $c_s(\mathbf{p}, \mathbf{q})$ is the spatial component defined as the Euclidean distance of $\mathbf{p}$ and $\mathbf{q}$'s pixel coordinates. The terms σ_c and σ_s are user-defined constants that control the spread of each term similar to the window size before.

The computation of the bilateral weights for each pixel in the input image is time consuming. A fast and qualitatively even better alternative to the bilateral weighting scheme in Eq.(8.5) is the guided image filter [He et al. 10].[3] While the output is similar to the bilateral weighting, the computation is different

$$w_g(\mathbf{p}, \mathbf{q}) = \frac{1}{|\mathbf{W}|} \sum_{\mathbf{k}:(\mathbf{p},\mathbf{q}) \in \mathbf{W_k}} (1 + (\mathbf{I}^l_{\mathbf{p}} - \mu_{\mathbf{k}})^\top (\Sigma_{\mathbf{k}} + \epsilon \mathbf{U})^{-1} (\mathbf{I}^l_{\mathbf{q}} - \mu_{\mathbf{k}})) \quad , \tag{8.6}$$

with $\mu_{\mathbf{k}}$ and $\Sigma_{\mathbf{k}}$ being the mean vector and covariance of $\mathbf{I}^l$ in a squared window $\mathbf{W_k}$ of user-defined size, centered at and being constant for each

[3] Code is available at http://research.microsoft.com/en-us/um/people/kahe/eccv10/



(a) (b)

Figure 8.3: 1D example of the guided image filter [He et al. 10] for a 1D signal. (a) The filter takes a guide signal (green) and fits it locally to the given, potentially noisy, cost function (blue) resulting in (b) a smoothed but edge preserving signal (red).

pixel $\mathbf{k}$. $|\mathbf{W}|$ denotes the number of pixels in the window. $\mathbf{U}$ is the identity matrix and ϵ a smoothness parameter. While, at first glance, Eq.(8.6) appears highly complex in comparison to Eq.(8.5), it turns out that the computation requires only running a series of box filters which can be computed in constant time, independent of the window size. For details see [He et al. 10].

Intuitively, the guided image filter takes a guide image, in this case the input image $\mathbf{I}^l$, and tries to fit it locally to the cost function $\mathbf{C}^d$ that encodes pixelwise costs for a certain disparity d.

$$\mathbf{C}^d_{\mathbf{p}} = c(\mathbf{p}, \mathbf{p} - d) = |\mathbf{I}^l_{\mathbf{p}} - \mathbf{I}^r_{\mathbf{p}-d}| \quad ,$$

This results in a smoothed version of $\mathbf{C}^d$ which is equal to aggregating the weighted costs in a given window around each pixel. For this, the best fitting linear transformation for local windows $\mathbf{W_k}$ around each pixel is computed, i.e., a scaling and an offset of the guide image, to get from $\mathbf{I}^l$ to $\mathbf{C}^d$. In a second step the linear transformation coefficients of all windows overlapping at a pixel are averaged. An example for a single-channel input is given in Figure 8.3. Other commonly used matching techniques and pixel descriptors can also be found in Chapter 7.

Cost volume filtering Stacking the functions $\mathbf{C}^d$ for all disparities d onto each other into a 3D array $\mathbf{C}$ creates the so-called cost-volume. The filtered cost-volume can be extracted by filtering each x, y-slice that belongs to a fixed disparity d with the guided image filter as described above. The final disparity for each pixel $\mathbf{p}$ is then defined in Winner-Takes-All manner as

$$d_{\mathbf{p}} = \underset{0 \leq d \leq d_{\max}}{\operatorname{argmin}} \mathbf{C}^d(\mathbf{p}) \quad .$$

Occlusion Occluded pixels are detected using a left-right cross checking procedure. Once the disparities for image $\mathbf{I}^l$ to image $\mathbf{I}^r$ are computed, one can exchange both images and additionally compute the disparities from $\mathbf{I}^r$ to $\mathbf{I}^l$. A pixel $\mathbf{p}$ is marked as invalid, i.e., occluded, if $d^l_{\mathbf{p}} \neq d^r_{\mathbf{p}-d_{\mathbf{p}}}$ where $d^l_{\mathbf{p}}$ is the disparity at pixel $\mathbf{p}$ with reference image $\mathbf{I}^l$. Again note that the disparity is always positive and the sign in Eq.(8.2) is changed according to whether the disparity for the left or right image is computed.

One cannot assign disparities to pixels being occluded in one of the input images. If the application demands such an assignment, it has to be based on some kind of sensible heuristic. In [Hosni et al. 11b] a weighted median filter is used for filling invalidated pixels.

Extensions An advantage of the presented framework is that it naturally extends to higher dimensional and more fine-grained solution spaces at the cost of higher computation times. In the previous example each slice in the cost volume corresponds to a certain integer-valued disparity. One can easily increase the precision to fractional values by increasing the number of slices and assigning each slice to a certain fractional disparity. More generally speaking, each slice of the cost volume can be considered to be a distinct label l from a set $\mathcal{L} = \{1, \ldots, L\}$. The user only needs to specify how these labels are mapped to semantically meaningful parameters for the algorithm. That means one is not bound to interpret l only as integer-valued disparities but could extend the label space to fractional disparities as well, e.g., [Gehrig et al. 12]. Alternatively, a set of slanted windows could be included to better handle slanted surfaces that are not fronto-parallel, e.g., [Gallup et al. 07].

By defining a mapping from a 2D solution vector (u, v) to the label space $\mathcal{L}$ one can directly extend the presented stereo approach to optical flow problems by exchanging $d_{\mathbf{p}}$ and d in Eq.(8.2) by $(u_{\mathbf{p}}, v_{\mathbf{p}})$ and (u, v), respectively. In this context it should be noted that modern optical flow methods mostly use a more sophisticated cost function including not only color- but also gradient-similarity; details for the presented approach can be found in [Hosni et al. 11b].

Limitations A principal limitation of all local methods, such as the one presented, is their inability to cope with highly ambiguous data such as unicolored walls or objects. Depending on the application this may not be crucial, e.g., for image interpolation, as no visible artifacts will occur if objects of the same color are interpolated incorrectly. In other applications, such as autonomous driving vehicles or robot navigation, this may pose a high risk, because there, accurate disparity, which means accurate depth, is crucial. Imagine similar stone pillars standing next to each other. Matching

the right ones is highly ambiguous. In such cases more complicated global correspondence estimation algorithms are required, a good overview is given in [Bleyer and Breiteneder 13].

Another limitation of the presented problem formulation is its discrete nature, which means it can only produce a solution that consists of combinations of preset labels. Even though labels may represent fractional values and the solution is therefore sub-pixel precise, it is always limited by the label space. The quality of any correspondence algorithm also depends highly on the scene content. While local methods are ranked high in the famous Middlebury benchmark [Scharstein and Szeliski 02], they are not always as successful in other benchmarks, e.g., [Geiger 12]. The reason could be a higher sensitivity to noise or ambiguities occurring more often in natural scenes. And finally occlusion handling can usually not be integrated into the matching process directly with local methods. Once the cost-volume has been created one could exchange the Winner-Takes-All strategy by a more sophisticated global label selection algorithm that could handle such cases by a better or more robust disparity assignment even for pixels occluded in one view. Therefore, the presented algorithm is a good starting point for further investigation of dense correspondence algorithms.

Section 8.4 extends the stereo correspondence estimation to multiple input cameras and deals with appropriate camera layouts and scene representations. Section 8.5 describes the plane-sweeping stereo algorithm that is easily portable to the graphics card to allow even real-time correspondence estimation.

8.4 Multi-View Stereo

The following section gives an overview of multi-view stereovision. The term multi-view stereovision (MVS) refers to stereovision-based reconstruction from $n > 2$ views, $\mathbf{I}^0$ to $\mathbf{I}^{n-1}$, and is sometimes called *multiocular stereovision* in contrast to *binocular stereovision* from one pair of views (Section 8.3). MVS has been an active field of research for several decades and more than seventy algorithms are listed on the Middlebury Multi-View Benchmark website [Seitz et al. 06]. This benchmark provides a commonly accepted test suite to evaluate the quality of multi-view stereo algorithms.

An important assumption of any MVS method lies in its required, compatible, or intended camera layout since various possibilities exist and may have an impact on the 3D reconstruction strategy (Section 2.2).

Most MVS methods (notably among those on the Middlebury list) are designed for n cameras freely laid out in space. Some apply binocular stereovision (as previously discussed in Section 8.3) on different couples of views

$(\mathbf{I}^i, \mathbf{I}^j)$ and then merge their separate binocular results. The main difficulty in such approaches concerns regularizing the union of separate results, especially in scene areas where reconstructions overlap. Common problems to be solved in such areas are to reduce too high point density and to resolve ambiguities/inconsistencies. This task pertains to point cloud merging and is discussed in Chapter 10 in more detail. Another type of MVS approach for a free camera layout consists in fitting some form of geometric model of the scene in order to maximize its local photo-consistency in available views [Furukawa and Ponce 10].

Some other MVS methods, sometimes called *multi-baseline stereovision* methods, are designed for the "parallel" or "decentered parallel" camera layouts discussed in Section 2.2 and especially fitted for 3DTV content capture. Those layouts are characterized by aligned, evenly distributed and parallel cameras. As will be demonstrated below, such restrained settings induce a set of geometrical constraints on corresponding pixels from different views arising from the so-called *simplified multi-epipolar geometry*. Those constraints enable searching correspondences over every view at once as a multi-view matching process [Okutomi and Kanade 93, Szeliski and Golland 99, Niquin et al. 10, Ismael et al. 14]. This is also called *multi-scopic stereo matching* and consists in matching n-tuples of pixels instead of couples which yields a consistent and more robust reconstruction.

In the following, the main concepts behind multi-baseline stereovision are reviewed. The "parallel" layout of Section 2.2 implies the optical centers $\mathbf{o}_0 \dots \mathbf{o}_{n-1}$ to be aligned and evenly distributed on the baseline, parallel optical axes orthogonal to this baseline, cameras of same focal and darkroom depth, sensors of same size and resolution $nc \times nl$ centered on their optical axes with rows parallel to the baseline. The "decentered parallel" layout (Figure 8.4), generalizes this setting by translating the sensor centers off their optical axis in such a way that the *line of sight* of all the views, defined for each camera by its sensor center and optical center, now converge on a chosen 3D *convergence point* $\mathbf{c}$, possibly at finite distance [Prévost et al. 13]. The convergence point is of utmost importance in 3DTV content shooting as it will be displayed exactly on the center of the 3D display and thus controls how captured scene space is mapped in the perceived 3D space (Section 2.2). One should note that, in this layout, the convergent lines of sight no longer coincide with the parallel optical axes. The off-axis translation of the "sensor" may be achieved both at hardware design stage as sensor chip physical/mechanical decentering and/or, to a given extent, at software post-processing stage as region of interest (ROI) cropping. In the following, the term sensor ROI is used to denote all of the above-mentioned possibilities.

Another benefit of such a layout is usually achieved thanks to rectification of the n views from aligned and evenly distributed cameras with

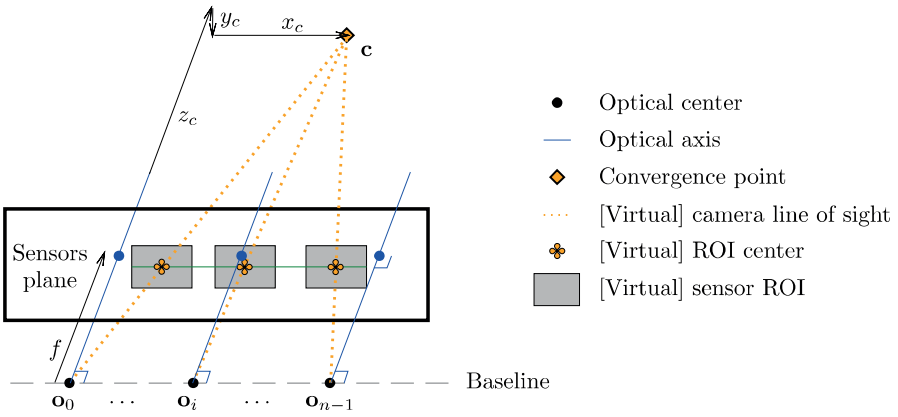


Figure 8.4: Decentered parallel camera layout.

(approximately) convergent optical axes. Similar to its binocular counterpart, the multiocular rectification consists in intersecting pixel rays by a plane at distance f (virtual sensors' plane) parallel to the common baseline connecting the optical centers (Figure 8.4). The rectified virtual sensor ROI in which the rectified views will be computed are then virtually laid in this sensors' plane with same size and orientation, so that the rows are parallel to the baseline. Furthermore, their centers are chosen to make every line of sight converge at the chosen 3D convergence point $\mathbf{c} = (x_\mathbf{c}, y_\mathbf{c}, z_\mathbf{c})$ (coordinates expressed in reference frame of camera 0) (Figure 8.4). One should note that there is not as much freedom in the layout of the actual cameras as in the binocular case as the rectification process relies on actual optical centers being aligned and rather evenly distributed.

For n images $\mathbf{I}^i$ recorded or rectified in "(decentered) parallel" layout and numbered $i \in \{0, n-1\}$ from left to right, the *epipolar constraint*, previously discussed for the binocular case (Figure 8.5), states that any pixel at $\mathbf{p}_i$ in any image $\mathbf{I}^i$ represents the actual 3D scene point projected onto $\mathbf{p}_i$. Pixel $\mathbf{p}_i$ and the optical center $\mathbf{o}_i$ of the camera are aligned on $\mathbf{p}_i$'s "pixel ray." Considering that pixel rays of corresponding pixels at $\mathbf{p}_i$ and $\mathbf{p}_j$ in two views $\mathbf{I}^i$, $\mathbf{I}^j$ have to intersect at their common 3D point $\mathbf{p}$, a straightforward derivation yields that optical centers $\mathbf{o}_i$, $\mathbf{o}_j$ and corresponding pixels at $\mathbf{p}_i$, $\mathbf{p}_j$ have to be coplanar (they belong to 2 intersecting and yet different lines). An *epipolar plane* is then defined for a couple of views (i, j) by both optical centers and any studied pixel in one of these views. The corresponding pixel in the other view has thus to be searched for within the *epipolar segment* defined as the intersection of this plane

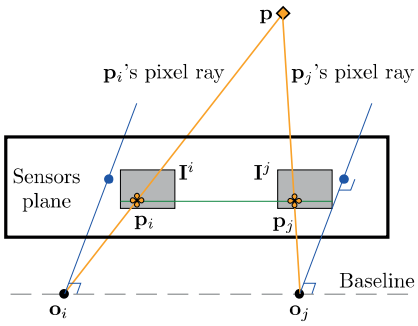


Figure 8.5: Simplified epipolar geometry.

with the other image (black horizontal line in Figure 8.5). When those two (rectified) cameras are set in "(decentered) parallel" layout, the epipolar segment in $\mathbf{I}^j$ defined by pixel $\mathbf{p}_i$ in $\mathbf{I}^i$ is part of the scanline of $\mathbf{I}^j$ of the same rank as the one holding $\mathbf{p}_i$ in $\mathbf{I}^i$.

In the multi-baseline context, because the optical centers are aligned, epipolar planes defined for a given pixel $\mathbf{p}_i$ in $\mathbf{I}^i$ and any other view $\mathbf{I}^j$ coincide. Successive pairwise binocular epipolar constraints thus ensure that corresponding pixels have the same y-value in every view $\mathbf{I}^i$. Hence, any 3D point $\mathbf{p} = (x_\mathbf{p}, y_\mathbf{p}, z_\mathbf{p})$ is projected into the [rectified] views $\mathbf{I}^i$, $\mathbf{I}^j$ onto corresponding pixels whose coordinates are respectively $\mathbf{p}_i$ and $\mathbf{p}_j = \mathbf{p}_i - (d^{i,j}_{\mathbf{p}_i}, 0)$, where $d^{i,j}_{\mathbf{p}_i} = u_i - u_j$ is called *horizontal disparity*.

A relationship between the horizontal disparity $d^{i,j}_{\mathbf{p}_i}$ and $\mathbf{p}$'s depth $z_\mathbf{p}$ can be established based on scale ratios between similar triangles. Let us consider the scale ratios in two pairs of such triangles, with apices on $\mathbf{c}$ and $\mathbf{p}$, respectively, and the camera centers $\mathbf{o}_i$ and $\mathbf{o}_j$ (Figure 8.6). Let $e_{i,j}$ be the distance between the center of the sensor ROI of camera i and j, defined by the triangle with apex on $\mathbf{c}$, and $e_{i,j} - d^{i,j}_{\mathbf{p}_i}$ be the distance between the corresponding pixels $\mathbf{p}_i$ and $\mathbf{p}_j$, in the sensors' plane, defined by the triangle with apex at $\mathbf{p}$. The relation between these two triangles yields the disparity-to-depth relation:

$$\left.\begin{array}{rl} e_{i,j} & = b_{i,j} \cdot (z_\mathbf{c} - f) \cdot z_\mathbf{c}^{-1} \\ e_{i,j} - d^{i,j}_{\mathbf{p}_i} & = b_{i,j} \cdot (z_\mathbf{p} - f) \cdot z_\mathbf{p}^{-1} \end{array}\right\} \Rightarrow d^{i,j}_{\mathbf{p}_i} = f \cdot b_{i,j} \cdot (z_\mathbf{p}^{-1} - z_\mathbf{c}^{-1}) \quad .$$

When the optical centers are evenly distributed (i.e., $\forall i, j \in \{0, \ldots, n-1\}, b_{i,j} = (j-i) \cdot b$), disparity values are scaled by $(j-i)$:

$$\forall i, j \in \{0, \ldots, n-1\} \quad d^{i,j}_{\mathbf{p}_i} = (j-i) \cdot f \cdot b \cdot (z_\mathbf{p}^{-1} - z_\mathbf{c}^{-1}) = (j-i) \cdot d_\mathbf{p} \quad .$$

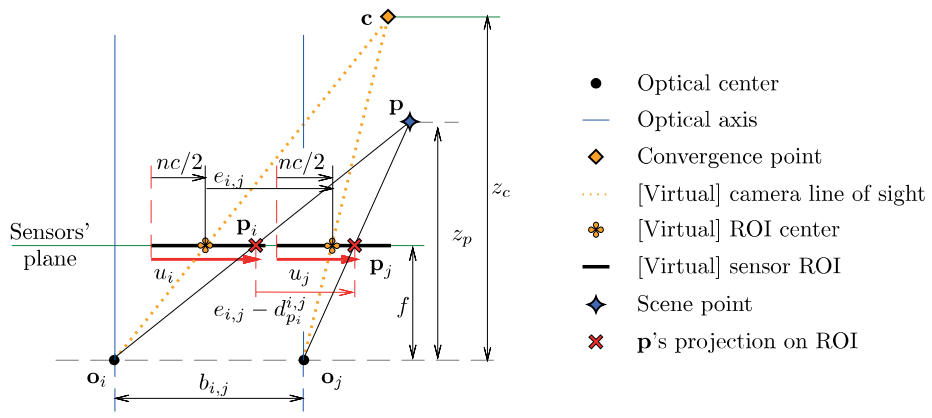


Figure 8.6: Projective geometry in off-axis simplified epipolar geometry (top view).

and disparity values for successive views are identical:

$$\forall i \in \{0, \ldots, n-2\} \quad d^{i,i+1}_{\mathbf{p}_i} = d_{\mathbf{p}}.$$

This common disparity $d_{\mathbf{p}}$ among each pair of successive views is conveniently used for each disparity assumption d for a pixel at $\mathbf{p}_i$ in $\mathbf{I}^i$. Instead of testing only two corresponding pixels for photo-consistency, one builds an associated geometrically consistent n-tuple in the multiscopic stereo matching process:

$$\forall j \in \{1, \ldots, n-1\} \quad \mathbf{p}_j = \mathbf{p}_i + (i-j) \cdot (d, 0) \quad . \tag{8.7}$$

These n-tuples contain one pixel per image, ordered according to their image number. Furthermore, thanks to Eq.(8.7), they all lie in the same epipolar plane and a common horizontal disparity assumption d is assigned to them. As such, pixels of a single n-tuple are corresponding projections on every view of a single 3D point.

To summarize, the presented multi-baseline stereovision paradigm reformulates the dense correspondence reconstruction problem as the answer to the question: "which of the geometrically consistent n-tuples correspond to actual 3D points in the scene according to their photo-consistency in the n views?".

Aside from the differences in the camera layout they are intended to handle, MVS methods may also be categorized according to what data or representation of the world they operate on [Seitz et al. 06]:

Scene-based methods employ a 3D scene model whose projections on views are checked for photo-consistency. As they are designed for a general freely arranged camera layout, they often use a 3D volume and rely on photometric similarity measures of the projections of the voxel cells, and remove others from the volume. Voxel coloring [Seitz and Dyer 99] preserves voxels whose cost is below a threshold. Space carving [Kutulakos and Seitz 00] progressively removes the photo-inconsistent voxels from an initial volume. More recently, a different category of methods has been proposed that uses a scene model composed of a collection of planar patches or surfels whose depth and orientation are separately optimized to maximize their photo-consistency. For representing patches, such methods rely on planar polygons [Habbecke and Kobbelt 06], circular disks [Habbecke and Kobbelt 07], or pre-segmented superpixels [Micusik and Kosecka 10]. The seminal work of [Furukawa and Ponce 10] fits patches on pixels around detected sparse features, then expands them in order to fill gaps between their projections, and afterwards reconstructs and refines a mesh.

Some multi-baseline methods make use of the disparity space introduced by [Yang et al. 93] for reconstruction instead of working on the standard 3D scene. Making use of photo-consistency and visibility reasoning, [Ismael et al. 14] optimize a so-called *materiality map* in this space for improved multi-view reconstruction.

Image-based methods compute a set of depth or disparity maps which are merged later [Narayanan et al. 98, Goesele et al. 06] or to which they apply constraints [Gargallo and Sturm 05, Szeliski 99] to ensure a consistent 3D scene reconstruction. Some methods that expect a more restrictive camera layout, typically multi-baseline, directly match n-tuples as multiscopic pixel sets [Niquin et al. 10, Kang and Szeliski 04], as described above. Among methods intended for a free camera layout, some computationally more intensive techniques are dedicated to MVS from community photo collections (CPC) and have gained an increasing interest. They have to handle a large number of uncalibrated views of a scene [Goesele et al. 07]. New difficulties then arise as such views are typically shot at different times, with differing acquisition geometries (viewpoints, angles, focal lengths, resolutions), and usually differing environmental conditions (weather, exposure, occlusions). This makes it necessary to restrict the matching to subsets of views sharing similar exposure, and empower the methods to deal with significant baselines (distances) between the cameras.

Feature-based methods compute dense correspondences by first matching feature points which can be more robustly estimated than a