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

Version Date: 20141204

International Standard Book Number-13: 978-1-4822-4382-6 (eBook - PDF)

Contents

Foreword ix

Preface xi

Contributors xv

Image Credits xix

The Editors xxiii

Acknowledgments xxv

I Acquiring the Real World 1

1 Camera Sensor Pipeline 3
Jan Kautz, Hendrik P.A. Lensch, Céline Loscos, and Philippe Bekaert
1.1 Introduction . . . 3
1.2 Sensor Technology . . . 3
1.3 Noise . . . 5
1.4 Demosaicing and Noise Reduction . . . 7
1.5 Radiometry and Color . . . 8
1.6 Geometric Calibration . . . 13
1.7 Summary . . . 21

2 Stereo and Multi-View Video 23
Laurent Lucas, Céline Loscos, Philippe Bekaert, and Adrian Hilton
2.1 Introduction . . . 23
2.2 Multi-Video Capture Geometry . . . 24
2.3 Calibration and Synchronization of Cameras . . . 28
2.4 3D Cameras in Practice . . . 36
2.5 Summary . . . 37

3 Omni-Directional Video 39
Peter Eisert and Philippe Bekaert
3.1 Introduction . . . 39
3.2 Mirror-Free Panoramic Capture . . . 40
3.3 Mirror-Based Panoramic Capture . . . 44
3.4 Stereoscopic Panoramic Capture . . . 46
3.5 Summary . . . 49

4 Range Imaging 51
Andreas Kolb and Fabrizio Pece
4.1 Introduction . . . 51
4.2 Structured Light Cameras—Kinect^{TM} . . . 52
4.3 Time-of-Flight Cameras . . . 56
4.4 Summary . . . 64

5 Plenoptic Cameras 65
Bastian Goldlücke, Oliver Klehm, Sven Wanner, and Elmar Eisemann
5.1 Introduction . . . 65
5.2 4D Light Field Acquisition . . . 66
5.3 Plenoptic Cameras . . . 67
5.4 4D Light Field Structure and Depth Reconstruction . . . 70
5.5 Spatial and Angular Super-Resolution . . . 71
5.6 Refocusing and Other Applications . . . 72
5.7 Summary . . . 77

6 Illumination and Light Transport 79
Martin Fuchs and Hendrik P.A. Lensch
6.1 Introduction . . . 79
6.2 Modeling Illumination . . . 79
6.3 Measuring Illumination . . . 80
6.4 Modeling Light Transport . . . 82
6.5 Measuring Light Transport . . . 85
6.6 Measuring Light Transport—Practical Issues . . . 86
6.7 Summary . . . 89

II Reconstruction—Data Processing Techniques 91

7 Camera Registration from Images and Video 93
Jan-Michael Frahm and Enrique Dunn
7.1 Introduction . . . 93
7.2 Structure from Motion Pipeline Overview . . . 94

7.3 Scalable Structure from Motion 107
7.4 Summary 109

8 Reconstruction of Dense Correspondences 111
Martin Eisemann, Jan-Michael Frahm, Yannick Remion, and Muhannad Ismaël
8.1 Introduction 111
8.2 Overview 112
8.3 Dense Correspondence Estimation 116
8.4 Multi-View Stereo 121
8.5 Stereo on the GPU 128
8.6 Summary 131

9 Sensor Fusion 133
Andreas Kolb, Jiejie Zhu, and Ruigang Yang
9.1 Introduction 133
9.2 Multi-Sample Fusion 134
9.3 Multi-Modal Fusion 142
9.4 Summary 150

10 Mesh Reconstruction from a Point Cloud 151
Tamy Boubekeur
10.1 Introduction 151
10.2 Overview 152
10.3 Registration 153
10.4 Outlier Removal 154
10.5 Normal Estimation 155
10.6 Point Set Surface 155
10.7 Meshing 158
10.8 Mesh Processing 159
10.9 Summary 160

11 Reconstruction of Human Motion 161
Yebin Liu, Juergen Gall, Céline Loscos, and Qionghai Dai
11.1 Introduction 161
11.2 Kinematic Skeleton and Skinning 165
11.3 Pose Optimization 168
11.4 Multi-Person Motion Capture 172
11.5 Capturing Hand Motion 175
11.6 Summary 176

12 Dynamic Geometry Reconstruction 179
Edmond Boyer, Adrian Hilton, and Céline Loscos
12.1 Introduction 179
12.2 Strategies 181
12.3 Sequential Shape Tracking 184
12.4 Non-Sequential Mesh Tracking 188
12.5 Motion Fields 190
12.6 Summary 192

III Modeling Reality 193

13 Rigging Captured Meshes 195
Kiran Varanasi and Edilson de Aguiar
13.1 Introduction 195
13.2 Overview 197
13.3 Fitting a Skeleton into a Static Mesh 199
13.4 Converting a Mesh Animation into a Skeletal Animation . 202
13.5 Building a Deformable Model 204
13.6 Summary 209

14 Statistical Human Body Modeling 211
Stefanie Wuhrer, Leonid German, and Bodo Rosenhahn
14.1 Introduction 211
14.2 Overview 212
14.3 Parameterization of Human Body Models 215
14.4 Statistical Analysis of Human Body Models 218
14.5 Summary 223

15 Cloth Modeling 225
Anna Hilsmann, Michael Stengel, and Lorenz Rogge
15.1 Introduction 225
15.2 Cloth Geometry and Mechanics Modeling 226
15.3 Cloth Deformation Modeling and Simulation 229
15.4 Cloth Appearance Modeling 232
15.5 Summary 238

16 Video-Based Character Animation 239
Dan Casas, Peng Huang, and Adrian Hilton
16.1 Introduction 239
16.2 Surface Motion Graphs 240
16.3 4D Parametric Motion Graphs 245
16.4 Summary 252

IV Authentic Rendering, Display, and Perception 253

17 Image- and Video-Based Rendering 255
Christian Lipski, Anna Hilsmann, Carsten Dachsbacher, and Martin Eisemann
17.1 Introduction 255
17.2 Plenoptic Approaches 255
17.3 Geometry-Assisted Approaches 261
17.4 Advanced Image-Based Methods and Extensions 268
17.5 Summary 274

18 Stereo 3D and Viewing Experience 275
Kai Ruhl
18.1 Introduction 275
18.2 Stereo Perception and the Human Visual System 276
18.3 3D Displays 278
18.4 3D Perception on 2D Displays 279
18.5 3D Video Representation 281
18.6 S3D Video Production from Real-World Data 283
18.7 S3D Delivery 288
18.8 Summary 289

19 Visual Quality Assessment 291
Holly Rushmeier
19.1 Introduction 291
19.2 Models from Human Perception 292
19.3 Experimental Techniques from Human Perception 294
19.4 Evaluation in Lighting and Material Modeling 295
19.5 Geometric Modeling 297
19.6 Motion 298
19.7 Image-Based Systems 299
19.8 Beyond Classic Models and Experimental Techniques . . . 300
19.9 Summary 301

V Applications 303

20 Facial Capture and Animation in Visual Effects 305
Darren Cosker, Peter Eisert, and Volker Helzle
20.1 Introduction 305
20.2 Static Facial Realism and Capture 305
20.3 Dynamic Facial Capture and Animation 308
20.4 Case Study: The Gathering 312
20.5 Summary 314

21 Television and Live Broadcasting 317
Graham Thomas, Philippe Bekaert, and Robert Dawes
21.1 Introduction 317
21.2 Sports Graphics 317
21.3 Challenges for Visual Computing in Sports Broadcasting . 318
21.4 Foreground Segmentation 320
21.5 Camera Calibration for Sports Events 321
21.6 3D Analysis 326
21.7 Virtual Broadcast Cameras and Second Screen 329
21.8 Summary 332

22 Web-Based Delivery of 3D Mesh Data 333
Max Limper, Johannes Behr, and Dieter W. Fellner
22.1 Introduction 333
22.2 3D Meshes vs. Image-Based Representations 333
22.3 Application Scenarios 336
22.4 Compression and Transmission 338
22.5 Summary 345

23 Virtual Production 347
Volker Helzle, Oliver Grau, and Thomas Knop
23.1 Introduction 347
23.2 Virtual Studios 348
23.3 Virtual Production for Cinema and TV 349
23.4 Real-Time Rendering with Game Engines 355
23.5 Summary 357

Bibliography 359

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

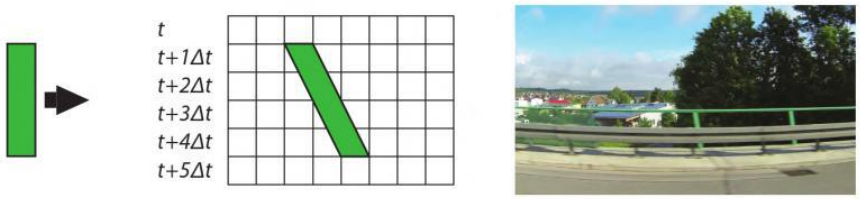

Figure 1.1: Rolling Shutter. In a rolling-shutter sensor each row will start exposure at a slightly different time. This results in distortion of moving parts. A vertical line moved to the right will be sheared.

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.

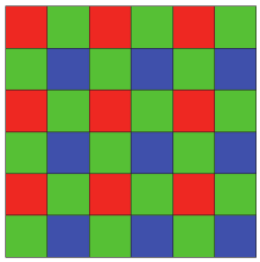

Figure 1.2: Example of the common Bayer color filter array (CFA).

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 sensitivies 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]:

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

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

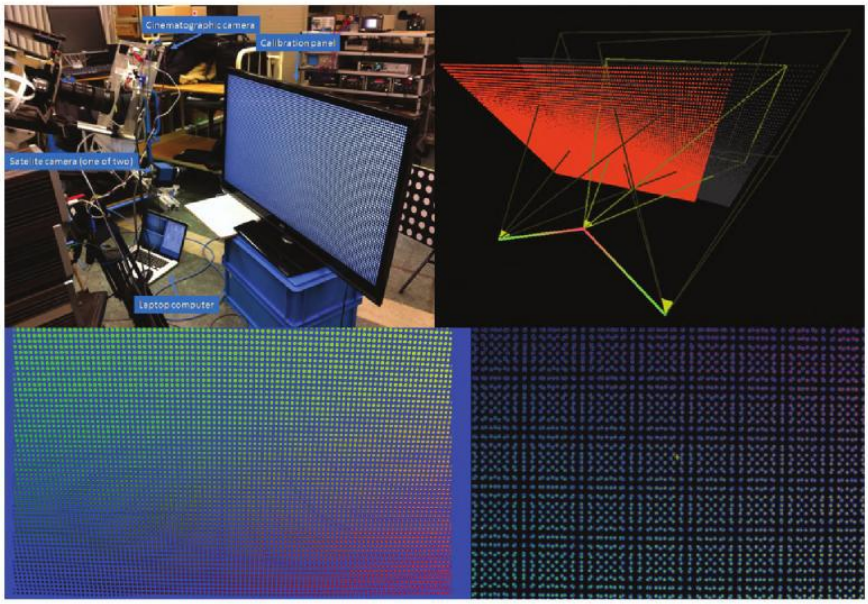

Figure 1.3: Trifocal camera rig calibration. Top-left: trifocal-camera set-up in front of computer-driven TV flat screen. Bottom-left: decoded calibration pattern on the central (broadcast) camera. Red corresponds with world-X, and green with world-Y coordinates. Note how lens distortion is significant, 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.