Technische Universität München
Lehrstuhl für Kommunikationsnetze
Fachgebiet Medientechnik

Acquisition and Streaming of Image-Based Scene Representations

Dipl.-Ing. Ingo Bauermann

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Klaus Diepold
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Eckehard Steinbach
	2.	Prof. Dr.-Ing. Marcus Magnor Technische Universität Braunschweig

Die Dissertation wurde am 20.11.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 30.01.2008 angenommen.

Acquisition and Streaming of Image-Based Scene Representations

Dipl.-Ing. Ingo Bauermann

April 11, 2008

Für Sabine und Marlene.

Acknowledgments

This thesis originates from my work as research scientist and teaching assistant at the Institute of Communication Networks at Munich University of Technology. During the time with the Media Technology Group, many people have contributed to this dissertation.

First of all, I owe gratitude to my supervisor Eckehard Steinbach not only for giving me advice and the opportunity to work on a fascinating topic, but also for the multitude of challenges beside my scientific work that I could meet and grow with. Professor Dr.-Ing. Eckehard Steinbach always supported me with his confidence, helpful discussions and ideas, and he gave me the liberty to develop and work out my own ideas.

I would also like to thank Prof. Dr.-Ing. Marcus Magnor for accepting to be the second auditor of my thesis and Prof. Dr.-Ing. Klaus Diepold for heading the committee.

I sincerely thank all my colleagues and students of the Institute of Communication Networks who have contributed to a friendly, creative, and stimulating atmosphere over all those years. Especially, I enjoyed the joint work and discussions with Yang Peng, Florian Schweiger, and Werner Maier. It always was a pleasure to consult other colleagues, namely, Peter Hinterseer and Julius Kammerl, and to work together with Professor Subhasis Chaudhuri.

My work with the Media Technology Group would not have been that enjoyable without the institute's administrative and technical staff. I would like to thank Dr.-Ing. Martin Maier, who has made life a lot easier, and Thomas Kurzhals for his help to build up the hardware I worked with. They and Sabine Strauss will always be remembered for their pleasant personalities and for their obliging manner.

I want to thank my parents for everything they did for me, especially, their willingness to allow me to take things apart, while knowing that I might not be able to put them back together.

Most of all, I wish to thank and to dedicate this dissertation to my wife Sabine for all of the patience and the love that she gives, and to my daughter Marlene for all the joy that she brings.

Munich, November 2007 *Ingo Bauermann*

Abstract

The common approach to three-dimensional computer graphics uses geometric scene descriptions. By simulating the interaction of light with objects, virtual views onto a scene can be generated. An alternative approach is image-based rendering. There, geometric scene descriptions are replaced by image-based scene representations and the simulation process is replaced by data interpolation.

One part of this thesis investigates the acquisition of unstructured image-based data sets using a multi-sensor platform and the virtual view generation from such data sets. In the second part the compression for interactive streaming of densely sampled image-based scene representations is investigated. Both topics cover key technologies for virtual reality applications like image-based 3D gaming or e-Commerce over the Internet.

The considered hand-held multi-sensor platform consists of three video cameras and a laser range finder. The acquired images and laser scans are registered in space and time to form an image-based scene representation. A new multi-sensor calibration procedure, pose estimation for several hundreds of images, and a sensor data fusion algorithm for 3D scene reconstruction are introduced. Real-time rendering from the images and their approximated local geometry under consideration of gross outliers during the preprocessing stages is performed. Objective quality metrics for such a system are evaluated and the timing behavior for interactive playout is examined.

The second part of this thesis deals with the compression and interactive streaming of densely sampled image-based scene representations. Objects and phenomenas like trees, glass, smoke, fog, and objects with complex properties like subsurface scattering, etc. can only be reproduced in a photorealistic manner from arbitrary viewing positions if a dense sampling of the scene is performed. Such scene representations do not suffer from outliers and inaccuracies during, e.g., the geometry reconstruction stage. But, due to the even higher amount of data that has to be processed, compression is inevitable. The most common video compression techniques provide high compression ratios and can be applied to image-based scene representations, but, these techniques do not allow us to access small parts of the compressed representation freely. Instead, a huge amount of image data has to be decoded to access the necessary parts for rendering. To overcome this drawback, in this thesis, a compression scheme is developed and evaluated that allows us to control the compression efficiency and at the same time provides real-time requirements that have an impact on the subjective feeling of interactivity of a rendering system. Considered constraints are the available channel throughput in a remote scenario and the computational capabilities of the client device. Both, theoretical and practical investigations are conducted and it is shown that the system's efficiency can be significantly improved with the proposed techniques. Finally, progressive rendering techniques are introduced that can be combined with such a compression and interactive streaming framework.

Kurzfassung

Gewöhnlich werden in der Computer-Grafik 3D-Bilder auf Basis von geometrischen Szenenbeschreibungen generiert. Die Interaktion von Licht und Objekten kann mittels solcher Beschreibungen simuliert und eine virtuelle Ansicht berechnet werden. Eine Alternative ist die Erzeugung von virtuellen Ansichten mit Hilfe von ausschließlich bildbasierten Szenenbeschreibungen. Die geometrische Szenenbeschreibung wird dabei durch Bilddatensätze ersetzt und die Bilderzeugung auf eine Interpolation von Bildelementen vereinfacht.

Ein Teil dieser Dissertation befasst sich mit der Aufnahme und Bilderzeugung von bildbasierten Szenenbeschreibungen. Dazu wird ein Aufnahmegerät untersucht, das aus mehreren unterschiedlichen Sensoren besteht. Ein zweiter Teil dieser Arbeit beschäftigt sich mit der interaktiven Übertragung von sehr dicht abgetasteten bildbasierten Szenenbeschreibungen. Beide Ansätze können beispielsweise in Bereichen der Virtuellen Realität oder der Spieleindustrie Anwendung finden.

Das Aufnahmegerät besteht aus drei Videokameras und einem Lasertiefenmesser. Die aufgenommenen Bilder und Laserdaten werden räumlich und örtlich registriert. Aus diesen Rohdaten lässt sich dann eine bildbasierte Szenenbeschreibung erstellen. Ein neuartiger Kalibrieralgorithmus wird vorgestellt, der eine metrische Rekonstruktion aller Abbildungsparameter erlaubt. Verfahren zur Positionsbestimmung des Aufnahmegeräts aus den aufgenommenen Daten und die Rekonstruktion der 3D Szenenstruktur werden untersucht. Die Berechnung und Darstellung von virtuellen Ansichten aus den gewonnenen Bildern und Tiefendaten unter Berücksichtigung von Fehlern in der Tiefenschätzung wird bezüglich dem Zeitverhalten des Systems und objektiver Qualitätsmaße evaluiert.

Ein zweiter Teil dieser Dissertation beschäftigt sich mit der Kompression und interaktiven Übertragung von dicht abgetasteten Bilddatensätzen. Phänomene wie Rauch und Nebel, komplexe Objekte wie Bäume und Gläser oder Oberflächeneigenschaften wie Spiegelungen lassen sich nur durch solche dicht abgetasteten Bilddatensätze modellieren. Auf diese Weise können photorealistische Ansichten erzeugt werden, ohne dass Fehler in der Tiefenschätzung Einfluß auf die Wiedergabequalität nehmen. Da aber eine sehr große Anzahl an Einzelbildern aufgenommen, verarbeitet und gespeichert werden muss, sind Datenkompressionsverfahren von großer Bedeutung. Üblicherweise werden Standardkompressionsverfahren verwendet, die sehr effizient unwichtige und redundante Datenanteile entfernen und hohe Kompressionsraten erzielen können. Um eine virtuelle Ansicht zu generieren, werden nur einzelne Teile des Gesamtdatensatzes benötigt. Standardverfahren erlauben es aber im Allgemeinen nicht, wahlfrei auf kleine Teile eines schon komprimierten Datensatzes zuzugreifen. Um dem entgegen zu wirken, wird ein Kompressionsverfahren vorgestellt und untersucht, das es erlaubt, sowohl die Kompressionseffizienz als auch die Realzeiteigenschaften, die maßgeblich durch den wahlfreien Zugriff bestimmt werden, in einem interaktiven System einzustellen. Die verfügbare Übertragungskapazität und die Rechenleistung des Empfängergerätes, die beide ausschlaggebend für den gefühlten Grad an Realismus sind, werden dabei in die klassische Raten-Verzerrungsoptimierung eingebunden.

Theoretische und praktische Untersuchungen zeigen, dass die Effizienz des Gesamtsystems dadurch signifikant verbessert werden kann. Abschließend werden progressive Übertragungstechniken vorgestellt, die zusammen mit dem zuvor beschriebenen Kompressionsalgorithmus einen Verzögerungs-Qualitäts Trade-Off während der interaktiven Übertragung erlauben.

Contents

List of Figures xv

List of Tables xix

Abbreviations and acronyms xxi

1 Introduction 1
- 1.1 Overview of the dissertation . . . 3
- 1.2 Contributions of the dissertation . . . 4

2 Background and related work 7
- 2.1 The plenoptic function . . . 7
- 2.2 Image-based rendering and scene geometry . . . 9
- 2.3 Acquisition of image-based scene representations . . . 11
  - 2.3.1 Structured scene representations . . . 11
  - 2.3.2 Unstructured scene representations . . . 14
  - 2.3.3 Hybrid systems . . . 16
- 2.4 Virtual view generation from image-based scene representations . . . 17
  - 2.4.1 Rendering from purely image-based scene representations . . . 17
  - 2.4.2 Rendering with explicit geometry . . . 18
- 2.5 Calibration and pose estimation . . . 19
  - 2.5.1 Single and multi-camera calibration . . . 20
  - 2.5.2 Heterogeneous multi-sensor calibration . . . 21
- 2.6 Geometry reconstruction for image-based scene representations . . . 21
- 2.7 Compression of image-based scene representations . . . 23
  - 2.7.1 Random access to samples of the plenoptic function . . . 23
  - 2.7.2 Intra-image compression . . . 24
  - 2.7.3 Inter-image compression . . . 24
- 2.8 Streaming of image-based scene representations . . . 26
- 2.9 Summary . . . 28

3 TRIVIS - Image-based rendering using a hand-held multi-sensor platform 33
- 3.1 Scene acquisition and rendering with TRIVIS - Overview . . . 33
- 3.2 Joint laser-camera calibration . . . 35
  - 3.2.1 Device setup . . . 35
  - 3.2.2 Sensor models . . . 36
  - 3.2.3 Calibration pattern . . . 37
  - 3.2.4 Initial solution . . . 37
  - 3.2.5 Refinement . . . 40
  - 3.2.6 Accuracy evaluation on synthetic data . . . 41
  - 3.2.7 Accuracy evaluation on real data . . . 44

3.3 Color calibration . . . 45
3.3.1 High dynamic range imaging . . . 46
3.3.2 Inter-camera color mapping . . . 47
3.4 Pose estimation using TRIVIS . . . 50
3.4.1 Overview . . . 50
3.4.2 Intra and inter shot feature matching and sparse reconstruction . . . 51
3.4.3 Direct pose estimation . . . 52
3.4.4 Feature tracking and merging for long sequences . . . 53
3.4.5 Global refinement . . . 55
3.4.6 Results . . . 56
3.5 Local scene geometry reconstruction . . . 59
3.5.1 Sensor data fusion . . . 62
3.5.2 Multi-view depth estimation for TRIVIS . . . 64
3.6 Robust rendering . . . 68
3.6.1 Blending multiple local models . . . 69
3.6.2 On the fly outlier removal . . . 69
3.6.3 Results . . . 70
3.7 Discussion . . . 71
3.8 Summary . . . 74

4 RDTC optimized compression - A theoretical analysis 77
4.1 System overview . . . 77
4.1.1 Compression using hybrid video coding concepts . . . 78
4.1.2 Interactive streaming . . . 80
4.1.3 System measures and parameters . . . 80
4.2 The decoding complexity . . . 83
4.2.1 Decoding a single block without a cache . . . 85
4.2.2 Decoding a single block with a pixel domain cache . . . 87
4.2.3 Decoding virtual views (with cache) . . . 89
4.3 The rate-distortion model . . . 93
4.4 A theory of RDTC optimal compression . . . 96
4.4.1 The RDTC model without a cache (Case I) . . . 96
4.4.2 The RDTC model with a bitstream cache (Case II) . . . 97
4.4.3 The RDTC model with a pixel domain cache (Case III) . . . 98
4.5 Issues with B pictures, 2D and hierarchical GOP structures . . . 101
4.6 Discussion . . . 104
4.6.1 Limitations and accuracy evaluation . . . 104
4.6.2 Lessons learned from the theoretical analysis . . . 105
4.6.3 Application . . . 106
4.7 Summary . . . 106

5 RDTC optimized compression - Practical coding 109
5.1 System overview of the practical RDTC coder . . . 109
5.2 Trained RDTC models . . . 110
5.2.1 A heuristic approach for RDTC optimization . . . 110
5.2.2 Trained models for RDTC system measures . . . 110
5.3 Optimization with respect to the initial delay . . . 114

5.3.1 Experimental results . . . 116
5.4 Optimization with respect to the mean delay . . . 119
5.4.1 Model parameterization for smooth navigation . . . 119
5.4.2 Experimental results . . . 120
5.5 RDTC optimization for interactive streaming . . . 121
5.5.1 RDTC optimization for concentric mosaics and line light fields . . . 122
5.5.2 Results for RDTC optimized compression of concentric mosaics . . . 123
5.5.3 Comparison to encoding with multiple representations . . . 123
5.6 Real-time experiments . . . 125
5.6.1 Overall system performance . . . 126
5.6.2 Impact of the caching system . . . 128
5.7 Decoding complexity constrained disparity compensation . . . 129
5.8 Discussion . . . 130
5.9 Summary . . . 132

6 Progressive rendering for RDTC optimized streams 135
6.1 Overview . . . 135
6.1.1 Multiple reference encoding . . . 136
6.1.2 Evaluation methodology . . . 138
6.2 Progressive transmission, decoding, and rendering . . . 140
6.2.1 Progressive interpolation . . . 140
6.2.2 Viewport resampling . . . 143
6.2.3 Image interleave . . . 145
6.2.4 Skip-scale progression . . . 147
6.2.5 Combining the progression schemes . . . 150
6.3 Discussion . . . 152
6.4 Summary . . . 152

7 Conclusion 155

A Appendix 159
A.1 Feature extraction and region matching . . . 159
A.2 Establishing correspondences for multi-sensor calibration . . . 160
A.3 Bundle adjustment . . . 160
A.4 Test data and error measures . . . 162
A.5 Evaluation methodology for streaming of structured representations . . . 163
A.6 Sampling of light fields . . . 163
A.7 Sampling of concentric mosaics . . . 165
A.8 The impact of the single reference block ratio on the decoding complexity . . . 166

Bibliography 169

List of Figures

1.1 Modeling and rendering approaches 2
1.2 Systems covered by this thesis 4

2.1 The 7D plenoptic sampling geometry and simplifications 8
2.2 The IBR continuum 9
2.3 Ghosting artifacts 13
2.4 The image-geometry space 15
2.5 Overview: Streaming of image-based scene representations 27

3.1 TRIVIS 34
3.2 TRIVIS: Device geometry 35
3.3 Error in the calibration parameters with respect to the noise level 43
3.4 Reprojection error after the initial estimate 44
3.5 Reprojection error after refinement 45
3.6 Reprojection error after initial and final estimate 46
3.7 Irradiance mapping of the cameras 48
3.8 Color mapping before and after calibration 49
3.9 Color matching between the cameras (on the calibration pattern) 50
3.10 Direct pose estimation for TRIVIS 53
3.11 Rematching vs. error propagation 54
3.12 Images of the test sequences 58
3.13 Registered cameras and point cloud 59
3.14 Feature-shot trackmatrix 59
3.15 Feature track length probability 60
3.16 Features per image 60
3.17 Reprojection error 61
3.18 Registered point cloud from the scanner data 64
3.19 View selection for multi-view stereo 66
3.20 View selection for multi-view stereo - five support views 66
3.21 Sensor data fusion results 67
3.22 Sensor data fusion results for TRIVIS 68
3.23 On-the-fly outlier removal 71
3.24 High dynamic range rendering with TRIVIS 72
3.25 Examplerendering 73

4.1 Block diagram of a hybrid video coder 78
4.2 GOP structure as used for video coding 79
4.3 Block diagram of the streaming system 81
4.4 Illustration of the single reference block ratio 83
4.5 Illustration of the decoding procedure 86
4.6 Single block decoding complexity without a cache 87

4.7 Statistical model for the decoding complexity with a cache 88
4.8 Decoding complexity of a single block request 89
4.9 Random access patterns 90
4.10 Decoding complexity for a frame access 91
4.11 Decoding complexity for a panoramic view access 92
4.12 Decoding complexity for arbitrary view access (beta=0) 93
4.13 Decoding complexity for arbitrary view access (beta=1) 94
4.14 Theoretical RDTC performance for single block access (no/bitstream cache) . 97
4.15 The caching gains with respect the decoding complexity without cache 98
4.16 Theoretical RDTC performance (single block; pixel domain caching) 99
4.17 Theoretical and practical RDTC performance for a single block access 99
4.18 Theoretical and practical RDTC performance for arbitrary virtual views . . . 101
4.19 1D hierarchical GOP structure (example) 102
4.20 2D hierarchical GOP structure (example) 104

5.1 The trained storage rate model 113
5.2 The trained distortion model 114
5.3 The trained transmission data rate model 115
5.4 Operational RDTC curves (RD) 117
5.5 Operational RDTC curves (RT and RC) 118
5.6 Access patterns for smooth navigation 119
5.7 The decoding complexity for second views 121
5.8 Operational RT and RC plots for smooth navigation 122
5.9 GOP structure for concentric mosaics 123
5.10 Operational RT and RC plots (joint optimization) 124
5.11 Block diagram of the streaming testbed 126
5.12 Testbed: Mean delay for random views 127
5.13 Testbed: Mean delay for rotation 127
5.14 Testbed: Mean delay for translation 128
5.15 Testbed: Mean delay for free motion 128
5.16 Testbed: Maximum delay for free motion 129
5.17 RC performance with decoding complexity constrained disparity compensation 131
5.18 Distribution of the disparity vectors under constraints 131

6.1 ANCHOR mode prediction 137
6.2 Progression: Considered navigation scenarios 139
6.3 Progressive interpolation: Visible artifacts 141
6.4 Progressive interpolation: Initial Delay&Standstill 142
6.5 Progressive interpolation: Rotation&Translation 143
6.6 Viewport resampling: Initial Delay&Standstill 144
6.7 Viewport resampling: Rotation&Translation 145
6.8 Image interleave: Initial Delay&Standstill 146
6.9 Image interleave: Rotation&Translation 147
6.10 Skip-scale progression 148
6.11 Skip-scale progression: Visual artifacts 149
6.12 Skip-scale progression: Initial Delay&Standstill 149
6.13 Skip-scale progression: Rotation&Translation 150

6.14 Combined progression: Initial Delay&Standstill 151
6.15 Combined progression: Rotation&Translation 151
6.16 Progressive Rendering: Visual artifacts 153

A.1 Input data for calibration of TRIVIS (first camera) 160
A.2 Input data for calibration of TRIVIS (laser scanner) 161
A.3 Concentric mosaic capture device 162
A.4 Captured images of the concentric mosaics test sets 163
A.5 Example renderings from concentric mosaics 164
A.6 Light field capture geometry 165
A.7 Concentric mosaic capture geometry 166
A.8 Impact of the single reference block ratio on the decoding complexity 167

List of Tables

2.1 Common image-based scene representations 9

3.1 Summary of the joint calibration algorithm 41
3.2 Reprojection error before and after bundle adjustment. 47
3.3 Numerical color calibration results 49
3.4 Summary of the pose estimation algorithm 56
3.5 Summary of the parameters and settings for the pose estimation algorithm . . 57
3.6 Reprojection error before and after global optimization (mean and standard deviation) for the 300 shot sequence. 61
3.7 Summary of the parameters and settings for the depth fusion algorithm . . . 69
3.8 Numerical results for rendering with outlier detection and removal. 72
3.9 Real-time performance for rendering from the scene representation acquired with TRIVIS. 72

4.1 Summary of encoding parameters and system measures 84
4.2 Decoding complexity for different block modes 84
4.3 Theoretical streaming performance of RDTC optimized scene representations 100

5.1 Results for RDTC optimized compression with respect to the initial delay . . 118
5.2 Performance of compressed representations for different rates 124

Abbreviations and acronyms

2D Two-Dimensional

3D Three-Dimensional

4CIF Common Intermediate Format (704x576 pixels)

bpp Bit per Pixel

ppp Pixel per Pixel

CG Computer Graphics

CIF Common Intermediate Format (352x288 pixels)

CV Computer Vision

DCT Discrete Cosine Transform

FGS Fine Granular Scalability

GBR Geometry-Based Rendering

GOP Group of Pictures

IBM Image-Based Modeling

IBR Image-Based Rendering

MPEG Moving Pictures Expert Group

MSE Mean Squared Error

PSNR Peak Signal to Noise Ratio

RANSAC Random Sample Consensus

RD Rate-Distortion

RDTC Rate-Distortion-Transmission Data Rate-Decoding Complexity

RDC Rate-Distortion-Decoding Complexity

RGB Red Green Blue

RMS Root Mean Square

SAD Sum of Absolute Differences

SfM Structure from Motion

SNR Signal to Noise Ratio

SSD Sum of Squared Differences

TC Transmission Data Rate-Decoding Complexity

VR Virtual Reality

YCbCr Luminance-Chrominance Color Space

ZNCC Zero-Mean Normalized Cross Correlation

1 Introduction

Images are an integral part of our life. The two images projected onto our retinas provide us with a huge amount of information. The interpretation by our **visual system**, i.e., by extracting basic elements like color, motion, orientation, and binocular disparity and subsequent further higher level processing, lets us build a three-dimensional (3D) model of our environment. The comparison of this model with the data provided by other senses, memories and experiences allows us to navigate in our environment and perform complex tasks. Beside the perception of our proximate environment we use technical equipment like cameras to capture, e.g., real photographs of a scene or event for **illustration**, **evidence**, **entertainment**, and **remembrance**. Images are used in many aspects of communication, especially in **visual communication**, to transport impressions and facts. Examples are movies, television, photography, print, computer games, art, and graphic design. Even **more general**, images can be anything that reproduces or approximates the appearance of some subject - from a drawing over simple water reflections to artificial images produced by computers.

Technically, and in the sense the word is used in this dissertation, an image is created by light rays falling onto a photo-sensitive surface. These light rays originate from a light source like the sun, interact with objects in the observed scene, and bundle in a single point in space, the center of projection. In this context images represent a set of **irradiance measurements** along a set of **viewing directions** captured at a specific **point in space**. Obviously, this concept resembles the image formation process by human eyes. And consequently, technical systems try to adapt the way the human brain processes visual data. A broad research field that addresses the design of visual sensors and image processing is called **Computer Vision (CV)**. One major topic found in computer vision research focuses on 3D scene reconstruction from real 2D photographs also referred to as **Image-Based Modeling (IBM)**. In addition, algorithms for, e.g., sensor calibration and pose estimation are investigated.

The applications of successful CV and IBM techniques are manifold. Autonomous, human like robot navigation is just one example. Another application is the generation of **virtual views** from a 3D scene description and belongs to the broad research field called **Computer Graphics (CG)**. Concepts developed by computer graphic scientists resemble the generation of mental images. A virtual view in this sense is an **image produced** from a more or less abstract description of an environment, a description consisting of, e.g., 3D objects, surface color, and relative positions as created using IBM techniques or by manual modeling. Such a view generation process based on geometric primitives is called **Geometry-Based Rendering (GBR)**. Additionally, the environment can be **augmented** with artificial or otherwise modeled objects. The main point here is that a virtual view has never been captured by real photographs, but is generated from the sparse and maybe manipulated description of a real environment at hand. Figure 1.1 shows the conventional processing pipeline for image-based modeling and rendering. From a set of photographs a 3D model is extracted using CV techniques like IBM. The model is then used to generate virtual views using GBR techniques like **texture mapping** and **warping**.

For complex scenes and objects that are **difficult to model** like hair, smoke, glass, and trees,

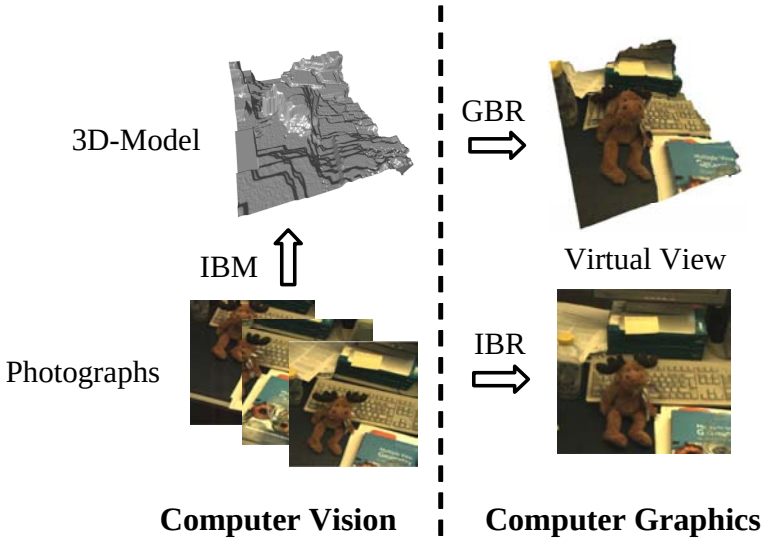

Figure 1.1: *Modeling and rendering approaches: Computer vision algorithms are used to extract camera parameters and a 3D model of the scene, e.g., by using image-based modeling techniques (IBM); View generation using computer graphics involve either geometry-based (GBR) or image-based rendering (IBR).*

the geometry-based modeling and rendering process can not achieve satisfactory results. An alternative rendering approach has emerged for such environments as also shown in Figure 1.1: **Image-Based Rendering (IBR)**. In IBR the conventional modeling process using IBM techniques and view generation using GBR techniques is replaced by a **direct processing** of the acquired image data. Conventional images captured at a sufficiently large number of viewpoints can then be used to achieve a photo realistic rendering.

For **interactive navigation** not only the rendering quality is relevant. The system **response time**, i.e., the time between virtual view selection and the display of that view, and the **frame rate**, i.e., the time between the display of two subsequent virtual views, are strong measures for the **subjective feeling of realism** of a walkthrough application. This becomes even more evident when the user has to download virtual views interactively over limited bitrate networks.

Fortunately, one of the main benefits of image-based rendering is the **scene complexity independent rendering** speed. In contrast, geometry-based rendering schemes might fail to render a virtual view within acceptable time whenever the scene becomes too complex, sometimes it is infeasible to use geometry-based rendering to produce photo realistic views. On the other hand, the main disadvantage of IBR is, of course, the need to acquire, store, transmit, and decode the **huge amount of image data** and to make image parts needed for rendering **accessible in real-time**. Again, in contrast, geometry-based rendering schemes only need a compact description of a scene to render arbitrarily chosen virtual views. To handle the memory usage and computational complexity issues of image-based rendering, efficient compression schemes adapted from **modern video standards** could be used. Unfortunately, random access to small parts of the image data **is not granted** with such coding schemes because of the exploitation of redundancy between acquired images. To maintain

both an efficient compression and at the same time free and fast access to image data, special coding schemes have to be developed.

In this thesis **two different systems** are considered, both of them providing all necessary techniques for the **acquisition and representation** of the underlying scene representation and **rendering** of virtual views at interactive rates.

The first system is based on a **multi-sensor platform** built of three video cameras and a laser range finder to support 3D scene reconstruction. The setup is chosen to improve the registration and scene reconstruction quality. This system uses a **hybrid approach** for modeling and rendering. I.e., the view generation incorporates both, image-based and geometry-based approaches. The main focus lies on **multi-sensor calibration**, **pose estimation**, and **3D scene reconstruction** for sparsely sampled scenes containing reflective and transparent objects.

The second system uses **image-based rendering approaches** solely. Here, the focus lies on **efficient compression and interactive streaming** of densely sampled representations. An optimized compression parameter selection procedure with respect to the system response time and frame rate is developed. The optimization **jointly considers** the storage rate, the distortion, and the available resources like **bitrate access** and **computational power** of the user's device. In this context a **theoretical framework** is presented and **practical issues** are addressed. Additionally, a real-time streaming **testbed** is implemented and progressive rendering techniques are evaluated.

1.1 Overview of the dissertation

This thesis covers a set of techniques belonging to different research areas. Figure 1.2 shows the processing blocks which are investigated in the following chapters.

For the first system, image and laser range data are acquired and system calibration is performed. The calibration procedures consider multi-camera color calibration including high dynamic range imaging, intrinsic calibration of the cameras, and relative translation and rotation of the cameras and the laser scanner. Pose estimation using the captured image data solely is performed considering the multi-sensor setup. Local scene structure is reconstructed using multi-view stereo techniques. Subsequent triangulation and geometry simplification follow to enable real-time rendering by the renderer that uses a reference view selection to determine the most appropriate subset of input images and their local geometry for rendering. The view selection performed by the renderer is fed back (offline) to the depth estimation stage to perform multi-view stereo using the cameras that are most probably selected by the renderer. This system can be used for the acquisition and rendering of scenes that contain very complex objects and at the same time needs only a relatively small number of input images.

For the second system, performing purely image-based streaming and rendering, calibrated image data is compressed using a new so called "RDTC optimization" approach which allows the adaptation to streaming scenario specific properties like available access bitrate and computational capabilities of the client device. The system covers the interactive bitstream assembly and cache synchronization to provide compressed image data to the decoder which in turn provides the decompressed image data to the renderer. The renderer performs view interpolation to display a requested view to the user. Additionally,

progressive rendering is performed driven by the renderer which tries to maintain a certain maximum system response time.

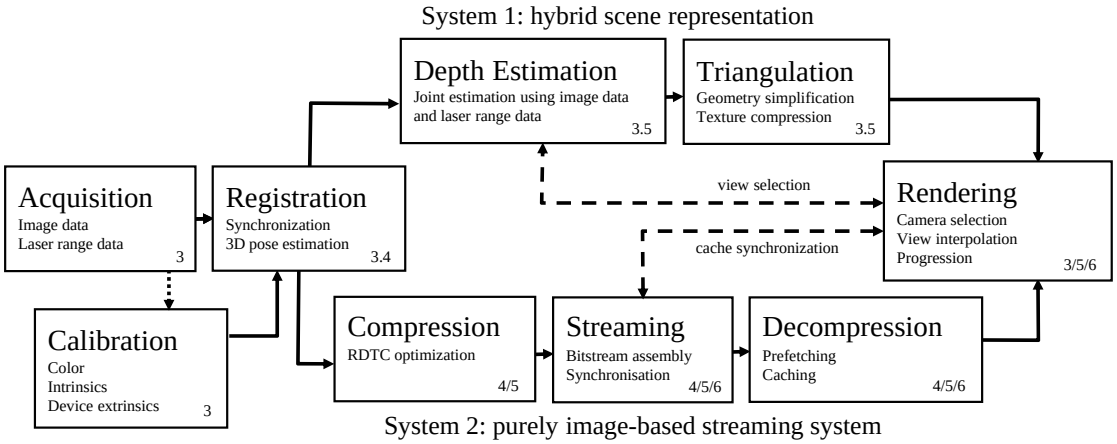

Figure 1.2: *Schematic overview of the processing blocks addressed in this thesis. The upper signal flow belongs to the hybrid image-based and geometry-based approach. The lower signal flow belongs to the purely image-based streaming system.*

The remainder of this thesis is structured as follows. In the next chapter the theoretical and practical background of image-based rendering and related work is discussed with respect to the application in later chapters. Common image-based scene representations are analyzed in detail. 3D scene reconstruction approaches as well as compression and streaming systems found in the literature are reviewed. Chapter 3 is dedicated to the calibration and acquisition techniques for the multi-sensor device. Algorithms for the joint estimation of the intrinsic and extrinsic parameters are presented and 3D scene reconstruction with respect to the properties of the acquisition device are given. Quality and timing metrics are evaluated for rendering from the acquired and registered image sequence. Chapters 4 and 5 study the compression for interactive streaming of purely image-based scene representations. A theoretical analysis is presented and a practical coder is evaluated. In Chapter 6 progressive rendering from compressed representations is discussed. This thesis concludes with Chapter 7 which gives a summary of the discovered insights and results.

1.2 Contributions of the dissertation

This thesis makes contributions in the field of acquisition and streaming of image-based scene representations. The main contributions are summarized as follows:

A multi-sensor acquisition device and its calibration

For the acquisition of unstructured light fields, a new hand-held multi-sensor platform is presented. The device consists of three video cameras and a laser range finder. A joint calibration procedure is developed and evaluated. The semi-automatic procedure results in a

full metric reconstruction of the physical sensor setup. Additionally, a pose estimation algorithm that is adapted to the multi-sensor platform is presented. The algorithm is designed to globally optimize the registration of a very large number of images.

Multi-sensor depth estimation and robust view interpolation from unstructured image sets

A multi-view depth estimation algorithm is presented that fuses image and laser range data to produce view dependent geometry. The reconstructed geometry is triangulated and used by a view generation procedure that is robust to outliers during the depth estimation and triangulation stages. The renderer is capable of displaying virtual views in real-time. The timing performance and quality measures for virtual view generation are evaluated.

Theory and practice for RDTC optimal compression and streaming

For the first time, theoretical models and a comprehensive analysis of rate-distortion optimal compression of densely sampled image-based scene representations considering scenario specific properties like available transmission bitrate and computational capabilities of the client device are presented. The analysis focuses on interactive streaming of precoded data in the context of densely sampled image-based scene representations. A practical framework for encoding parameter estimation is also given and a comparative evaluation is worked out to show that such a system can significantly reduce the user perceived delay during online operation compared to common approaches.

Progressive rendering techniques for interactive streaming of image-based scene representations

Progressive rendering techniques for interactive streaming of densely sampled image-based scene representations are investigated. Four techniques working with a single stream that has been encoded using the RDTC framework are compared according to their delay vs. distortion performance. An evaluation with respect to independent and rate-distortion optimal encoding based on disparity compensation is also given. The progression schemes can be combined and do not sacrifice the compression efficiency.

A testbed for streaming of image-based scene representations over the Internet

A streaming testbed is implemented and used to evaluate the real-time behavior of the RDTC optimal compression and streaming scheme as well as for the progressive transmission techniques. A synchronized caching system is used and the impact of finite size client side caching on the real-time behavior is investigated.

2 Background and related work

In this chapter an overview of image-based rendering techniques is given. The goal is to provide the reader with the basic understanding of image-based rendering, the corresponding scene representations, and their acquisition methods. Rather than presenting a comprehensive study, this chapter works out the fundamental properties of image-based modeling and rendering techniques with respect to the subsequent chapters. In the later sections, sensor calibration, compression and streaming approaches related to image-based modeling and rendering found in the recent literature are discussed. Comprehensive studies can be found in [SKC03, SCK07, Mag05].

2.1 The plenoptic function

The fundamental concept that all IBR representations have in common is the plenoptic function [AB91]. The plenoptic function is a seven dimensional function that defines the visual appearance of a scene completely. In its most general form it describes the intensity of "light rays passing through any point in space":

$$P(X, Y, Z, \psi, \varphi, \lambda_l, t). \tag{2.1}$$

Given the parameterization of this function for a specific scene, the intensity value for every light ray is registered by its viewpoint (X, Y, Z), direction (ψ, φ), wavelength λ_l and time t. Figure 2.1 (left) shows the sampling geometry for one light ray leaving a light source, hitting a scene object and being captured. Generating views is just a matter of composing appropriate intensity values of rays passing through the desired center of projection of the virtual camera. The goal of every image-based rendering technique must be the full reconstruction of the plenoptic function to provide the possibility to generate every imaginable view onto a scene. This is a challenging task mainly because of the high dimensional signal processing involved and the fact that it is practically infeasible to sample the plenoptic function at its minimum sampling rate for real scenes. To overcome these practical shortcomings several simplified versions of Equation (2.1) have been proposed.

A rather theoretical concept, the surface plenoptic function as defined in [Zha04], is an example for a representation in six dimensions. Here, it is assumed that the intensity of light rays does not change along the path of propagation. In this way, a 2D parameterization of the scene's surface is used to specify the sample point position rather than three dimensional Euclidean coordinates. This reduction by one degree of freedom leads to a six dimensional representation. Due to the properties of the human visual perception, the wavelength λ_l can be discretized to red, green, and blue (RGB) for computer graphics applications (this simplification can be made due to the phenomenon of metamerism [Wys58, Wan95]). Some image-based rendering techniques additionally assume a static scene (time t has no longer to be considered). An example is 'Plenoptic Modeling' introduced in [MG95] where panoramic images (2D) are registered in space (3D) to end up with a 5D scene representation. The

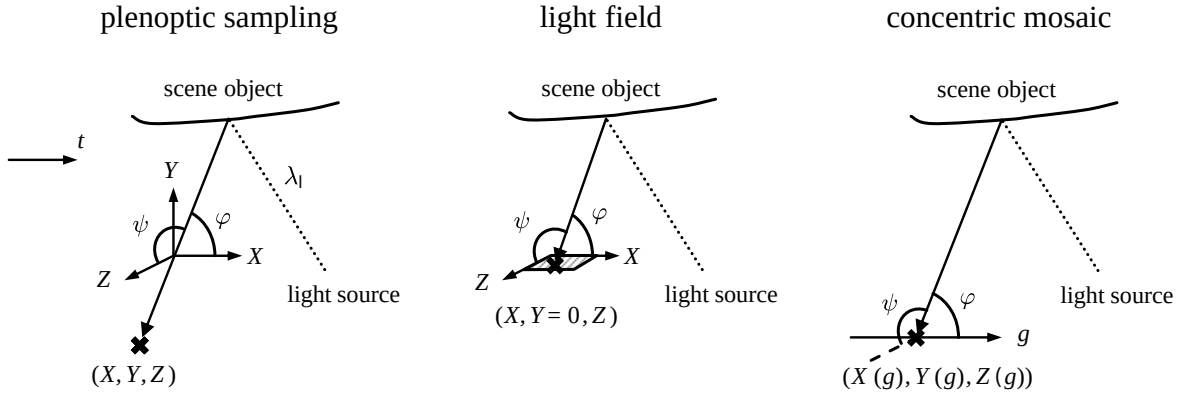

Figure 2.1: *The 7D representation for full plenoptic sampling (left), the reduced 4D parameterization of light fields (middle), and the reduced 3D representation for concentric mosaics (right).*

most popular scene representations are the Light Field [LH96] and the Lumigraph [GGSC96, BBM+01]. Again, both representations reduce the dimensionality of the underlying scene description by the assumption that the intensity of a light ray does not change until it is blocked by the scene. This allows a parameterization with only two spatial coordinates (e.g., Y is no longer considered). Together with the propagation direction (in the plane-sphere representation like in [CLF98]) of a light ray, these scene representations utilize a 4D parameterization of the viewing space. Figure 2.1 also illustrates the sampling geometry for this light field approach (middle). The simplified plenoptic function becomes:

$$P(X, Z, \psi, \varphi). \tag{2.2}$$

A further simplification of the plenoptic function, namely removing degrees of freedom for the position and orientation of a virtual camera, leads to another popular scene representation: concentric mosaics [SH99]. For this representation the user movement is restricted to a circular area. Beside the impractical way of capturing slit images on concentric circles, concentric mosaics are commonly captured using an outward looking camera at the end of the beam of a camera crane. During rotation of the crane, the camera captures images at different positions on a circle. In this way, the position of a light ray sample can be parameterized by a single variable (g in Figure 2.1 (right)). Additionally, the propagation direction of the light ray is captured by two angles resulting in a 3D parameterization. A related representation but with a completely different kind of possible user navigation are "movie maps" introduced in [Lip80]. Movie maps are image sequences captured along a certain path and single frames can be chosen and displayed according to the desired viewing location. Several of these videos are concatenated to assemble a large environment to virtually navigate in.

For ordinary images or panoramas the plenoptic function degrades to 2D. Apple's Quicktime VR [Che95] is one example, using 360 degree cylindrical panoramic images that are captured at discrete points in a scene. A user can choose such a discrete viewpoint and the viewing direction.

An overview of the introduced image-based scene representations and their number of dimensions is given in Table 2.1. The large number of possible ways to sample the plenoptic

function led to a variety of scene representations in the past 15 years. Most of them rely on additional geometric information to reduce the number of images required as discussed in the next section.

dimensions	example representation	selected references
7D	plenoptic function	[AB91]
6D	surface plenoptic function	[Zha04]
5D	plenoptic modeling	[MG95]
4D	light field/(unstructured) Lumigraph	[LH96],[GGSC96],[BBM+01]
3D	concentric mosaics/movie maps	[SH99],[Lip80]
2D	panorama/image	[Che95]

Table 2.1: *Common image-based scene representations*

2.2 Image-based rendering and scene geometry

Image-based scene representations can be classified according to the degree of geometry information they utilize for rendering. The location of some common techniques in the IBR continuum is shown in Figure 2.2. On the left hand side, scene representations using no or only approximated geometry can be found where also those described in the former section reside. Toward the right hand side, more geometry information is incorporated into the representation as it is done in classical geometry-based modeling and rendering approaches.

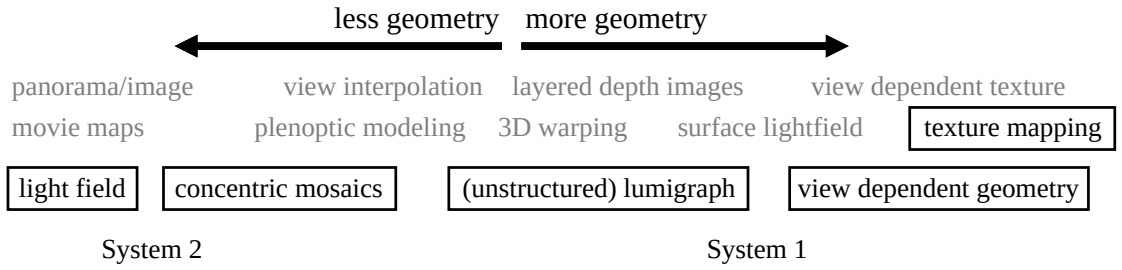

Figure 2.2: *The IBR continuum (according to [SCK07]).*

For images, panoramas, and movie maps no geometric information is used at all to produce photorealistic renderings. The same applies to the classical light field approach [LH96] where a huge amount of images is captured to avoid undersampling of the plenoptic function. For rendering with concentric mosaics it is simply assumed that the scene is planar and resides at a constant distance from the capturing cameras.

Plenoptic modeling belongs to the class of representations utilizing implicit geometry information. Implicit geometry information does not use descriptions like 3D primitives (points, lines or polygons in 3D). Instead, point correspondences in multiple images are established manually or automatically from the set of input images themselves. Then, a dense field

of image coordinate displacements derived from the point correspondences is extracted by interpolation. Intensity values from the source images are interpolated at their estimated position in the novel views. Further representatives of this class are the view interpolation methods introduced in [CW93, LF94] and the view morphing approach in [SD96]. A recent approach working full automatically has been presented in [SL05]. Implicit geometry is most often used with uncalibrated or weakly calibrated image sets. Please note that the definition of implicit geometry may differ from author to author in the literature. However, the main disadvantage of implicit geometry is that a virtual camera can not be placed within a common world frame due to the lack of explicit geometry information.

This disadvantage can be resolved with fully calibrated image sets. In this third class of IBR representations, explicit information about the scene structure is used which also has to be provided as side information along with the input images. The geometric information is automatically (see Section 2.6 on page 21) or manually extracted from the images themselves or provided by additional sensors like laser scanners (e.g., [CL96]).

The surface plenoptic function [Zha04] relies on accurate geometry information. For all points on the parameterized surface a bunch of rays is captured and stored. 3D warping (e.g., [McM99]) can be used to generate a new virtual view onto a scene from a single or multiple images when pixel wise depth information is available. Layered-depth images [SGHS98, CBL99] provide multiple depth values and corresponding intensities for each pixel location in an image. When a novel viewpoint is chosen near to the original camera position, occlusion and parallax are rendered correctly by warping the input pixels from back to front into the virtual view.

Texture-mapping is one of the fundamental geometry-based rendering approaches using images. Vertices of a 3D mesh or another parameterization of the scene geometry are associated with image coordinates. The vertices are projected into a desired view and the corresponding image coordinates are used to interpolate the input image to produce a novel view. View-dependent texture mapping techniques [DTM96, PCD+97, DBY98] generate novel views in a similar way. But, instead of using a single image as a texture, multiple images and image coordinates that depend on the current viewpoint and viewing direction are used for mapping onto one common scene geometry description. This allows the best image with respect to the current virtual viewpoint to be selected as a texture for the common 3D model. View-dependent geometry techniques go a step further by also allowing the geometry to be view dependent. Originally designed for artificial scenes [Rad99], this approach becomes attractive especially with very complex scenes where automatic 3D scene reconstruction is very challenging or a scene can not be modeled by a single common geometric description [ESK03, ESK05].

The Lumigraph, as an image-based scene representation closely related to the light field approach, exhibits a strong dependency on the geometry used. Although rendering can be performed without geometry, it is used for resampling unstructured input data to a structured representation (i.e., a light field). Unstructured Lumigraph rendering [BBM+01] does not perform the resampling step and therefore needs geometric information to be included into the scene representation for proper rendering. Unstructured Lumigraph rendering unifies view dependent texture mapping and conventional light field rendering.

When scanning the literature on image-based modeling and rendering techniques, one basic principle emerges: The more geometry is utilized, the less images are needed. For scenes

with a complex illumination and fine details, generally more images are required to ensure a high quality rendering. Some approaches are not feasible for real scenes, others put restrictions on the user movement that might not be tolerable.

Representations that are used in this thesis are analyzed in more detail in the next sections and are marked with a bounding box in Figure 2.2. System 1 corresponds to the system discussed mainly in Chapter 3 using view dependent geometry, texture mapping, and view interpolation similar to unstructured Lumigraph rendering [BBM+01]. System 2 is purely image-based and used for interactive streaming in Chapter 4, 5, and 6.

2.3 Acquisition of image-based scene representations

Data acquisition for image-based rendering systems is an ongoing research area. Acquisition devices and appropriate rendering techniques are coupled and are discussed frequently in the literature addressing critical sampling of the plenoptic function. Undersampling results in visible artifacts while oversampling wastes storage and may decrease the rendering performance for interactive systems. Generally, optimal sampling of Equation (2.1) is influenced by three factors:

- the complexity of the texture of the scene,
- the scene structure, and
- the desired rendering resolution.

In order to determine the minimum sampling rate for aliasing free virtual view generation, several theoretical and practical approaches have been investigated. Generally, these approaches can be categorized in structured and unstructured approaches.

2.3.1 Structured scene representations

Structured representations acquire and store images in such a way that the access to a single light ray can be simply derived from the regular sampling pattern. Most structured scene representations do not use explicit geometry information. Light fields and concentric mosaics are two of the most common representatives.

Light Fields

Light fields can be acquired by placing cameras on a regular 2D grid with the optical axes perpendicular to the grid surface. Also placing the camera on a hemisphere with the camera inward looking at an object is very common. For the former case, sampling of its four dimensional representation has been studied using a geometrical [LS00] and optical [CCST00] analysis. A light field representation is critically sampled if the maximum displacement of scene objects in adjacent images is smaller than one pixel. Under the assumption that no occlusions occur and only diffuse materials are present in the scene, the minimum spacing of capturing cameras ΔX_{max} needed to avoid aliasing artifacts during rendering can be determined by

$$\Delta X_{max} = 2 \cdot \frac{d}{f_c} \cdot \frac{z_{max} \cdot z_{min}}{z_{max} - z_{min}} \tag{2.3}$$

if the focal plane of the virtual camera is placed in a distance of z_{opt} with respect to the plane where the *capturing* cameras reside:

$$z_{opt} = 2 \cdot \frac{z_{max} \cdot z_{min}}{z_{max} + z_{min}}. \tag{2.4}$$

Where d is the pixel diameter, f_c represents the focal length of the capturing cameras, and z_{min} and z_{max} are the minimum and maximum distances of the scene from the cameras, respectively. An illustration of the sampling geometry with a constant depth assumption is given in Appendix A.6 on page 163. The desired resolution of a virtual view is directly involved with the choice of the pixel size d. The scene's texture complexity does not have an impact on the number of images that have to be acquired if the output resolution is assumed to be the input resolution of the capturing cameras (e.g., d=1 pixel).

It has been shown in [ZC03b] and [DMMV05] that for non-Lambertian (real scenes are non-Lambertian in general due to, e.g., surface reflections and refractions) no minimum sampling rate exists. For practical systems including slightly differing parameterizations from the original, e.g., spherical light fields [IPL97], Equation (2.3) is still a good approximation [ZC03b]. By taking the limited resolution of the capturing devices into account, this expression ensures that when the virtual camera simultaneously faces the objects nearest and farthest from the capturing cameras, no aliasing artifacts appear under the assumption that the scene resides at a constant depth at z_{opt}.

In [ZC03b] it has been shown that regular sampling of images on a grid as proposed in the original work on light fields [LH96] is not the most compact representation. In [ZC03b] quincunx or hexagonal sampling lattices are proposed that reduce the sample rate significantly. This is achieved by nesting the fan like spectra in an optimal way. A drawback is that complex filters have to be used to reconstruct virtual views making this approach infeasible in practice.

Looking at Equation (2.3) reveals the fact that for general scenes the spacing between adjacent images on a plane grid have to be that close to each other that the lenses would touch. Indeed, some practical systems do exactly this. Such a light field camera has been build in [NLB+05] by inserting a microlens array between the sensor and the main lens. That way, subimages at different locations on the camera's image plane can be captured and their pixels can be rearranged to produce effects like varying depth-of-field from a single shot. A similar approach is the light field microscope in [LNA+06]. Virtual navigation is, of course, only possible in a very limited fashion.

Instead of simultaneously capturing images, moving cameras may be used. For viewing of small objects the light field gantry from [LH96] consists of a turntable where the object can be placed on. The capturing camera can be translated and rotated so that it faces the object during rotation of the turn table. From the images acquired from various spatial positions and viewing angles, a light field parameterization is extracted. This principle setup and procedure is often used with minor changes like in [IMG99], [WAA+00], or [MPZ+02].

Large camera arrays have been built to capture scenes with background rather than single objects. An example is the Stanford high performance light field array using 256 cameras that can be configured in many ways [WSLH02, WJV+05]. Though the applications are manifold for such large camera arrays, the acquisition of light field representations is one of them. A 64 camera setup has been presented in [YEBM02]. Even dynamic light fields can be captured with these devices. A self configurable array that is capable of moving 48 capturing cameras

according to the estimated scene structure to avoid undersampling of the light field has been built in [ZC07].

Due to the huge amount of data that has to be acquired, the light field representation is not commonly used for large scale interactive navigation applications. Restricting the cameras to lie on a single line in space is applicable in some cases and is often called a line light field. Such a special case are concentric mosaics which are discussed in the next paragraph.

Concentric Mosaics

Concentric mosaics [SH99] are captured on a curved line. Usually, a rotating camera crane is used with an outward looking video camera attached to the end of the beam. Following the reasoning in [ZC03b], under the assumption that no occlusions occur and only diffuse materials are present in the scene, the maximum rotation angle $\Delta\xi_{max}$ of the camera crane between the capture of two images needed to avoid aliasing artifacts during rendering can be approximated by

$$\Delta\xi_{max} = 2 \cdot \frac{d}{f_c} \cdot \left(\frac{z_{min}}{z_{min} - R} - \frac{z_{max}}{z_{max} - R} \right)^{-1} \tag{2.5}$$

if the focal plane of the virtual camera is placed in a constant distance of z_{opt} with respect to the center of the concentric mosaic:

$$z_{opt} = R \cdot \left(1 - 2 \cdot \left(\frac{z_{min}}{z_{min} - R} + \frac{z_{max}}{z_{max} - R} \right)^{-1} \right)^{-1} \tag{2.6}$$

Where R is the radius of the camera path. d is the pixel diameter, f_c represents the focal length of the capturing cameras, and z_{min} and z_{max} are the minimum and maximum distance of the scene from the center of the concentric mosaic, respectively. Equations (2.5) and (2.6) are derived in Appendix A.7 on page 165. To illustrate the ghosting artifacts that occur from undersampling, Figure 2.3 shows a detail of rendered views from one of the compressed test sequences used in later chapters at different sampling rates. In the top image part the sampling rate and constant depth is chosen according to Equations (2.5) and (2.6). 1525 images are captured in 4CIF resolution with $R = 1.5m$, $\frac{d}{f_c} = \frac{1}{750}$, $z_{min} = 3.5m$, and $z_{max} = 20m$. The constant depth computes as $z_{opt} = 5.1m$. From top to bottom the sampling rate is halved from one figure part to the next. The image part in the bottom of Figure 2.3 is undersampled by a factor of 16. Significant aliasing artifacts can be observed in horizontal direction in the undersampled cases.

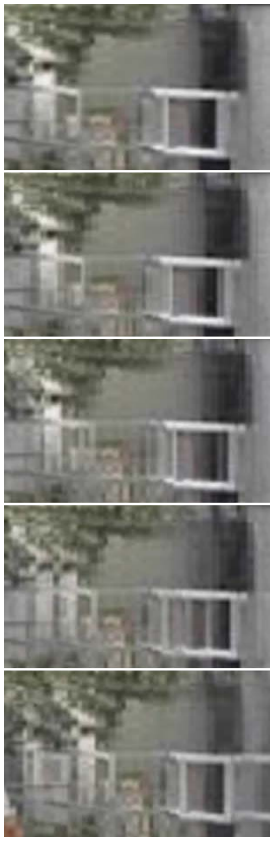

Figure 2.3: *One detail of a scene rendered using concentric mosaics. The sampling rate decreases from top (critically sampled) to bottom (16 times subsampled).*

However, the main drawback with concentric mosaics is that no matter how dense it is sampled, there are vertical distortions when the virtual camera is off the capturing path of the camera. Without more accurate knowledge of the scene structure these distortions can not be compensated for. In the original work [SH99] a depth correction scheme was presented. Nevertheless, in

case of complex scenes, perfect reconstruction is not possible in general. Moreover, due to the vertical distortions, a concatenation of concentric mosaics to achieve a large area to freely place a virtual camera in, is not possible. The main advantage of concentric mosaics is the simple acquisition procedure. A camera array may be used instead of a single camera to capture a light field-like representation with a curved surface where the capturing cameras lie on. Such a device has been proposed in [LZWS00], but no real world results have been presented. Approximated uniform acquisition from a hand-held camera has been presented in [CKS00] and [PBE99] where the latter uses concentric mosaics to produce stereo panoramas.

2.3.2 Unstructured scene representations

Unstructured scene representations do not constrain the capturing cameras to be placed on a regular grid or evenly spaced on a line. Instead, the captured images are registered most often by additionally providing information about the acquisition structure like per view projection matrices which contain information about the intrinsic parameters of the capturing cameras as well as their pose. For structured acquisition, theoretical bounds can be derived as shown in the previous section. For unstructured sampling not only the number of images has to be determined, but, also their optimal placement is unknown which makes a general analysis challenging.

Geometry adaptive light field sampling

Although light fields are structured representations as introduced in [LH96], known scene structure can have a great impact on the number of images that have to be acquired. Then, one can perform an adaptation of the sampling pattern to the scene structure which has to be provided as side information along with the images themselves. In [CCST00] a minimum sampling curve was proposed that jointly considers the number of input images and the accuracy of the used scene geometry. The main idea is that in the occlusion free case and when only diffuse scene objects are present, the scene can be split into objects at different depth layers. With other words, multiple constant depth assumptions are made and for every depth layer a subscene can be independently acquired and rendered. For each of the layers at depth z_i the minimum spacing of the according capturing cameras can be determined from Equation (2.3). The depth distribution of depth layers assuming a given number of layers N_D and the minimum and maximum depth of the scene z_{min} and z_{max} can be determined as follows:

$$\frac{1}{z_i} = \lambda_i \cdot \frac{1}{z_{min}} + (1 - \lambda_i) \cdot \frac{1}{z_{max}} \text{ where } \lambda_i = \frac{i - 0.5}{N_D}. \tag{2.7}$$

Note that Equations (2.3) and (2.7) formulate a N_D vs. ΔX_{max} trade-off. Figure 2.4 shows two applications of the minimum image-geometry space analysis. The minimum number of images with respect to the accuracy of the 3D model is shown (minimum sampling rate). Given a fixed number of images the minimum accuracy is indicated. Vice versa, for a fixed accuracy of the 3D model, the minimum number of images is denoted. Sampling points above the curve are redundant.

Another way to incorporate geometry information into the acquisition of light fields is to realize that objects at the minimum and maximum depths in a scene are unlikely to be visible

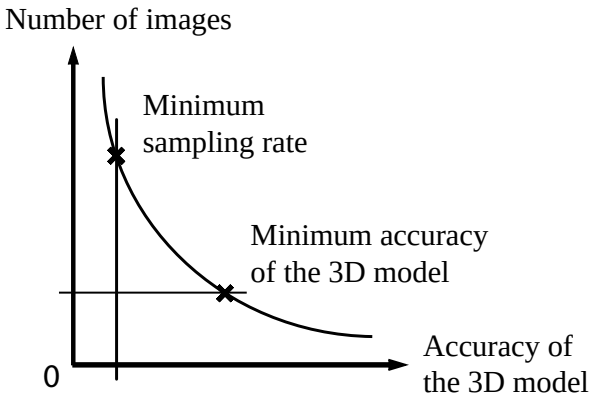

Figure 2.4: *The image-geometry space from [CCST00]).*

from every camera. Then the scene is split into several smaller scenes with much smaller depth variation than the complete scene. Due to the fact that an adaptation to the scene geometry is performed in the cases mentioned in this section, in this thesis light fields using scene structure for rendering are considered as unstructured representations.

Non-uniform sampling

Theoretically, for known scene geometry, the minimum sampling rate can be determined using (2.3) and (2.7). However, in practice and for the case of known scene structure, automatic camera placement is performed mostly using heuristics. In [FCOL00, TTV+02] tentative camera placements within a predefined viewing space are evaluated using the scene model and are ranked by visibility and quality. In [VFSH04] an objective measure called viewpoint entropy is proposed that considers the visibility of all scene objects from a certain position of a virtual camera to rank possible new viewpoints to capture input images from. Both approaches then select a small subset of views as the final scene representation. In [SHS99] an adaptive algorithm for the acquisition of synthetic scenes is proposed. A per view error measure is defined and minimized within a predefined viewing space. The algorithm emphasizes the reconstruction quality because the error measure is directly derived from the rendering process. A mesh representation of the viewing space where each vertex represents a captured image is refined until the overall error measure is minimized or other requirements are met.

Also not guaranteeing critical sampling, a couple of approaches can be found in the literature for unstructured acquisition without prior knowledge of the scene. In the original work on Lumigraph rendering [GGSC96] a hand-held camera was used. Aided by visual markers in the scene, the cameras' pose are recovered and used to resample the acquired images to a structured light field representation. In [HPDvG99] a system is described that also performs acquisition from a hand-held camera. Intrinsic and extrinsic calibration is performed on a set of images facing the scene. From the acquired images the pose of the cameras are recovered using image correspondences. Virtual views are generated from a subset of the input images by using interpolation on a coarse mesh of point correspondences. A system which dynamically decides if a captured image is inserted into the scene representation is introduced in

[LSG02]. The measure is based on the absolute difference between the reconstruction of a single light ray using the actual state of the representation and the corresponding acquired sample, starting from a manually selected set of images. In [ZC03a] a scheme is proposed that uses a coarse to fine strategy for inward looking concentric mosaics. The reconstruction error for in-between views is estimated based on a color consistency criterion starting with a small set of widely spaced images. The next best position of the camera is chosen according to the estimated maximum error to ensure to capture the image which has the highest impact on the overall distortion. In [AC01, AFYC03] an initial path planning is done by hand for a manually placed vehicle with an omnidirectional camera mounted on an eye-height plane. The huge amount of image data (up to ten thousands of images for large scale environments) is registered from image correspondences. The extracted feature points also serve as implicit geometry approximation for reconstruction through interpolation of three warped views. Another system for the acquisition of large scale environments is reported in [ESK05] using a multi-sensor rig built of four cameras. Again, image registration is performed from the images themselves. A depth map is calculated for every captured image and is later used for virtual view interpolation using view dependent geometry and texturing. In [ZC07] a light field capturing device with 48 cameras mounted on a plane surface is set up that can rearrange the cameras on a plane in order to optimize the visual appearance of reconstructed virtual views. The system iteratively minimizes a scene geometry dependent error measure using vector quantization techniques with respect to a virtual viewpoint on the fly.

2.3.3 Hybrid systems

Especially for large scale scenes, several systems using more than one of the basic image-based scene representations can be found in the literature. In [KWLS03] and [BS03] it was proposed to divide a large and complex real world scene into smaller parts that can be represented by different sampling methods. Stair cases and narrow hall ways in real scenes can be efficiently modeled, e.g., using panoramic videos instead of higher dimensional representations when the restrictions to the viewing space are acceptable, e.g., for applications like virtual walkthroughs. For rooms, e.g., concentric mosaics could be used, whereas object models with even higher detail could be inserted using light fields. The concatenation of different scene representations is very challenging which is mainly due to distortions like those observed for concentric mosaics. In [HCTL02] the manually aided insertion of light fields into panoramic images is investigated. A city walkthrough application using panoramic video with view dependent texture mapping is proposed in, e.g., [KIS01].

Augmented reality applications which combine artificial geometry-based and image-based real world objects are possible when reliable geometry information is available. The gaming application in [BS04b] might serve as an example. A laser scanner is used for depth acquisition for a panoramic view into which animated 2D image objects are inserted. Additionally, stereo perception is realized for a single panorama resulting in a very low complex gaming application where the user can freely rotate and interact within the environment. A more sophisticated system like in [TAL+07] uses model-based geometry and motion reconstruction of a human actor from multiple video cameras. Additionally, reflectance characteristics are captured which can be used to place the actor in an artificial or real world environment with realistic illumination in real-time while choosing a virtual viewpoint onto the hybrid scene.

2.4 Virtual view generation from image-based scene representations

The captured intensity samples and depth information, if provided, form the scene representation and serve as the basis for the reconstruction of the plenoptic function in Equation (2.1). Virtual view generation typically consists of collecting relevant samples from the reference data and appropriate filtering (e.g., interpolation). Both steps mainly depend on:

- the intensity sample density,
- the camera calibration accuracy, and
- the accuracy of the depth information.

Due to practical limitations most scene representations are undersampled. Nevertheless, ad hoc computation or prior knowledge of the scene structure allows photorealistic rendering in many cases.

2.4.1 Rendering from purely image-based scene representations

If a geometry model is not available, basic ray interpolation as proposed in the original work on light field rendering can be performed [LH96]. There, a low pass filter is applied to the image samples to ensure that no aliasing artifacts appear in the final rendered image. The filter operation actually means that the input images are downsampled which resembles the findings of rendering with a constant depth assumption while placing the constant depth at the focal length of the acquisition cameras. Better results can be obtained when the camera resolution is kept as is and a constant depth for view interpolation during rendering is chosen as described in the analysis for light field and concentric mosaic sampling (compare to Section 2.3.1 on page 11). This reconstruction method assumes that only Lambertian surfaces are present in the scene and no occlusions happen. Light rays that are to be reconstructed are usually approximated by four samples from cameras nearest to the virtual view (or the light ray) that hit the constant depth plane close to the point where the considered virtual ray hits that plane. Practically speaking, the improvement in quality when sampling and reconstructing using the constant depth assumption (compared to the original light field rendering approach in [LH96] with identical sample density) is that ghosting and blurring (like in Figure 2.3) is minimized as the maximum disparity in adjacent input images is bounded within +/-1 pixels. The constant depth assumption is suitable for representations that do not contain too much reflections and other details. From heavily undersampled representations this method produces blurred views.

The wide aperture approach in [IMG99] interpolates many light rays to reconstruct a single pixel in a virtual view. A clear appearance of the scene can only be achieved for objects at a narrow depth interval. Details far from the chosen depth appear blurred. Moreover, specular highlights are completely lost due to averaging. By using many cameras far apart from each other a wide synthetic aperture for the virtual camera is simulated.

In [SYGM03] the previous mentioned approaches are combined to produce views that are sharp at a certain adjustable depth but do not produce too much blurring in regions of the virtual view containing features at other depths. This is obtained by rendering with a constant depth, low-pass filtering and then adding details from the wide aperture approach. Nevertheless, for heavily undersampled light fields and in animation this reconstruction

method has advantages over rendering with a constant depth assumption as it produces less temporal inconsistent ghosting effects. Similarly, in [TKN03, KTAC04] a virtual view is rendered at different depths. A focus measure is optimized on the produced virtual base images to find weighting factors for each of them. The final rendered image is the weighted average of the "multiple focus" base images. A theoretical justification is given in [TN06].

To ensure that the output view appears approximately as sharp as the input images even when rendering from heavily undersampled image sets, in [FWZ03] a scheme is presented that uses image priors. Image priors are a set of pixel patches in the original images from which one should be similar to every output patch. Especially in animation, inconsistent patches are visible due to the global per view optimization. Real-time depth estimation as performed in [LCM+06, CLM07, ZC07] share the same restrictions as view dependent geometry to support pixel interpolation is calculated on the fly by using a local color consistency constraint.

The rendering complexity is fairly low for quadrilinear reconstruction based on the constant depth assumption. For the other approaches a significantly larger complexity is observed.

2.4.2 Rendering with explicit geometry

If a 3D geometric model is included in the scene representation, view generation can be performed by projecting pixels from captured images into 3D-space and by reprojecting them onto the image plane of the virtual camera. If the geometry is accurate, only few images allow for high quality rendering of simple scenes provided that only Lambertian surfaces are present in the scene. Geometry information can be available in a variety of different formats. Explicit geometry is often represented as per pixel depth or a 3D mesh. 3D meshes are a collection of 3D vertices connected by edges to form geometric primitives like triangles, quadrangles and other polygons. Also the intrinsic parameters as well as the position and orientation of the capturing cameras have to be provided with the images.

In the simple form of rendering geometric primitives with texture, each vertex is associated with a texture coordinate. The texture of the interior of a triangle or polygon is defined by the interior of the triangle or polygon that the texture coordinates surround in a captured image. During rendering, a projective mapping of the model's vertices into the desired virtual view is performed. Once the projections of the vertices in the virtual view are known, an interpolation of the intensity values at interpolated texture coordinates of visible polygons is performed on a per pixel basis in the virtual view to form the output image. A Z-buffer ensures that occlusions are considered during the warping process. Principally, only one single image (texture) and the scene geometry have to be captured or generated to render a simple scene. Because the geometry as well as the texture remain static, effects like reflections can not be reproduced.

Usually, for image-based rendering multiple views onto the scene are available. In this case, view dependent texture mapping is an appropriate tool [DBY98, DTM+98] and works just like texture mapping except that each model vertex can be associated with texture coordinates from multiple images. In this way, a texture can be chosen or blended from multiple images that have been captured most closely to the viewpoint of the virtual camera. With this technique one can achieve viewpoint dependent rendering of details like specular

highlights, semi-transparent surfaces, etc. The number of images that have to be provided along with the scene geometry depends on the complexity of the scene and can be dramatically reduced compared to rendering with purely image-based scene representations. This is exploited when only a sparse subset of the available images can be processed due to performance reasons. Under the assumption that the scene is Lambertian, the image-geometry space analysis in Section 2.3.1 on page 11 can be adopted by realizing that the provided scene model can be interpreted as a single layer with varying depth. In theory, the number of images needed depends on the accuracy of the geometric model. In practice, non-Lambertian surfaces and occlusions have to be taken into account. A pop-up light field [SSY+04] is such a representation using view dependent texture mapping and manually aided geometry extraction.

One step further, even the geometry can be view dependent. Especially, for systems that extract the geometry from the input images themselves, depth estimation errors are very likely. These errors show up as annoying ghosting artifacts and become worse with the distance of the virtual camera to the reference images. For complex objects in [PCD+97] an approach for rendering with view dependent geometry is proposed that uses structured light for scene reconstruction. Different approaches including rendering with view dependent geometry from a hand-held camera have been compared in [HPDvG99] showing that view dependent geometry performs very well compared to view reconstruction assuming a constant depth or a global 3D model. In [ZKU+04] a system using eight fixed cameras for view reconstruction from a dynamic scene is presented that also makes use of view dependent geometry in conjunction with an alpha blending algorithm to enhance the rendering quality near object boundaries. In [ESK05] the acquisition and rendering of large scale scenes with a camera rig is considered. As no reliable depth information can be extracted for very complex scenes, one common 3D model can not be computed. Instead, they use local 3D models that allow the reconstruction of the plenoptic function from a few capturing cameras near the virtual view based on the local geometry model that is assumed to be valid at the position of the virtual camera. In this way, depth estimation errors do not propagate very far. As a drawback, temporal inconsistent view dependent models are produced that cause artifacts during animation and user movement.

Rendering from unstructured images has been, in general, investigated in [BBM+01]. The rendering process in this work is a generalization of multiple techniques. It allows for rendering from a set of images as it is done for structured representations when no or only approximate geometry is available (e.g., [GGSC96, SCG97]). For unstructured representations providing geometry information, the algorithm behaves like view-dependent texture mapping (e.g., [ESNK06, DBY98]). Common graphics hardware can be efficiently used for rendering of very complex geometric models making some of the above mentioned approaches suitable for real-time rendering.

2.5 Calibration and pose estimation

As mentioned in the former section, beside the sample density, a major factor for high quality image-based rendering is the camera calibration accuracy. This involves both the intrinsic parameters of the capturing cameras and their pose. Intrinsic parameters like the focal length, pixel skew, the principal point, and lens distortions can be recovered from image

projections of known scene points. Such algorithms have been well studied even for special cameras [WCH92, GD99, Zha00] and software tools are freely available [Bou07, SFA07]. Pose estimation algorithms have been proposed for visual sensor systems as well as for heterogeneous systems consisting of cameras and laser range finders.

2.5.1 Single and multi-camera calibration

A lot of work has been done to determine the pose of a single camera from image sequences. Structure from motion techniques work directly on point correspondences (extracted using, e.g., [Can86, HS88, SS90, Low04]) or line correspondences (e.g., [TK95, BS05]) in multiple images. Fully automatic approaches often determine intrinsic and extrinsic parameters jointly [HZ04]. For large still image data sets such a system is presented in [SSS06]. The early work on Lumigraph rendering used markers in the scene to register the single capturing camera in 3D-space [GGSC96]. The original light field system in [LH96] used a single camera and a special gantry which allowed to register captured frames accurately. Such robot arms, camera arrays or camera cranes [SH99] are often used to acquire images without the need to register the acquired data from the images themselves or to support automatic registration.

Also for multi-camera systems structure from motion techniques exist where the intrinsic camera parameters and at the same time the relative positions of the cameras with respect to a device origin are estimated. In [SMP05] a calibration system is proposed that works on simultaneously acquired images of a laser dot from at least three cameras in a darkened environment. Point correspondences are automatically extracted and a metric camera model up to a scale is calculated by projective factorization [ST96] and bundle adjustment (e.g., [TMHF99]). The fixed acquisition geometry and intrinsic parameters for larger camera arrays are often recovered by the use of calibration patterns (e.g., [VWJL04]).

For image-based scene representations the spacing of adjacent images is usually very small when captured from a moving camera. Moreover, for the acquisition from mobile devices, standard methods for sequential pose estimation suffer from error propagation when tracking over long image sequences solely using image features. If a single camera is used to acquire an image-based scene representation, images that are spatially near but are acquired at significantly different time instances might not be well calibrated. In [KPG00] this problem is tackled and a system is presented that allows to accurately calibrate an image sequence which was obtained by moving the acquisition camera in serpentines in front of an object. Spatially near frames are detected and registered jointly to avoid error propagation. Pose estimation for a multi-camera system used for image-based modeling and rendering is described in [FKK04]. Assuming that the intrinsic calibration of the single cameras is available, the algorithm estimates their relative pose directly from acquired scene points using standard structure from motion techniques [HZ04]. For sequences showing largely untextured areas or for a camera movement with heavy rotation of the capturing device during acquisition, the feature tracking used with these systems usually fail even when robust image features like SIFT-features [Low04] are used.

Individual cameras in multi-camera systems almost always have slightly different intensity and color mapping characteristics. Though both intrinsic calibration and pose estimation are most often robust against such deviations, during later processing steps (e.g., geometry reconstruction and view blending) they have a great impact. Color calibration solves for the radiance to intensity mapping for every camera or at least minimizes the mismatch between

cameras with respect to one arbitrarily chosen reference camera. For camera arrays such a procedure has been described in [JWV+05, IW05]. These methods first linearize the sensor respond function by manipulating the camera's hardware settings and then further minimize the inter-camera color mismatch in software. To obtain high dynamic range images from a set of multi-exposure images in [DM97] an approach is presented.

2.5.2 Heterogeneous multi-sensor calibration

To acquire explicit geometry information along with the images, some systems use dedicated hardware especially for large scale scenes and unstructured acquisition. Per pixel depth values or mesh-based geometry representations can be extracted from line laser range finders (e.g., [SIC07]) or 2D laser image scanner (e.g., [RIE07]). Laser scanners have their main application in robotics for applications like self localization, quality management, etc. In image-based modeling they are often used for geometry reconstruction and visualization using texture mapping techniques (e.g., [EHBR98, LPC+00, Dia03]). For such purposes the scanner device and the camera have to be calibrated relative to each other. For the calibration of a single camera to a line laser scanner (only capturing depth values on one specific plane in space), a method using a 3D calibration pattern has been presented in [ZP04a]. For 2D laser scanners (providing depth images) and a camera, similar approaches can be found in the literature (e.g., [UH05]).

Sensor pose estimation is supported by a laser range finder in [FZ02] for planar movement. In [BS05] a technique for the joint calibration of a moving camera-scanner system for the acquisition of concentric mosaics has been proposed which incorporates both relative sensor pose estimation and pose estimation for the joint camera-scanner device.

2.6 Geometry reconstruction for image-based scene representations

As mentioned earlier, also the accuracy of the geometric model has an impact on the reconstruction quality that can be achieved. Beside the acquisition of geometry with dedicated hardware as described in the former section, scene structure can also be computed from the acquired images themselves. For a comprehensive survey on stereo reconstruction from two images see [SS02], for reconstruction from multiple images see [SCD+06]. Generally, for image-based rendering two different scene setups are considered in the literature. In the object centered approach modeling and rendering of a single object of interest is considered. The user should be able to look at, and freely move around, the object. The view centered approach on the other hand tries to visualize a scene from arbitrary viewing positions lying within and facing outside a specific viewing space. In the following some basic geometry reconstruction techniques are reviewed with respect to the object centered and view centered scene setups.

Voxel-based reconstruction divides the working volume into 3D-voxels. Each voxel is labeled as opaque or transparent by voting from multiple pixels in the input images. The obtained 3D model can be meshed or converted to depth maps by reprojecting the voxels into the source images. Occlusions in the scene are handled by the order voxels are traversed during reconstruction and taking account of voxels that are already considered opaque and

therefore occlude other voxels. Such algorithms have been presented in [SD97, ESG99, BC00, ESG00, SCMS04, VBK05] and are most commonly used for object centered scenes.

Silhouette-based reconstruction is similar to voxel-based reconstruction in the sense that both approaches carve away the space where the object does not occlude the background. The silhouette of an object is extracted in many views and the visual hull [Lau94] of an object is constructed from these silhouettes. The main difference to the voxel-based approach is that for silhouette based methods only the object's convex hull is extracted. A considerable advantage is the processing speed for such methods. Typical algorithms can be found in [MBR+00, CTMS03, MM05, FLB06]. Silhouette-based methods are suitable for object-centered but not for view-centered representations as a foreground-background separation is often not possible.

Pixel-based reconstruction is typically carried out by setting up a correlation volume which consists of a set of correlation measures (e.g., normalized cross correlation, sum of absolute differences etc.; see Appendix A.1 on page 159) at discrete disparity hypotheses for every pixel in a reference view with respect to one or more matching images. For two-view stereo images the correlation can be measured for shifts along epipolar lines [LH81]. For multiple baseline stereo a plane sweep can be performed [Col96] to obtain the correlation volume. Local optimization may be performed to find the best matching disparity or depth for each pixel separately, but ambiguities due to untextured areas lead to noise in the reconstruction. To avoid this, the correlation may be measured on windows surrounding the pixel under consideration. An alternative is global optimization by considering that neighboring pixels are likely to have a similar disparity (known as the smoothness constraint). Such global optimization schemes are dynamic programming (e.g., [OK85]), the graph cut (e.g., [KZ02]) and belief propagation (e.g., [SZS03, SLKS05]) or heuristics like in [Hir06, Hir07]. Once a per pixel disparity map is determined, the scene structure is obtained by triangulation (e.g., [HZ04]). Pixel-based methods are general purpose methods that work on both object centered and view centered representations. The computational complexity and memory consumption might be high when global optimization is performed so that the image resolution is limited in practice.

Color segmentation is a further general purpose approach to reduce matching ambiguities and to handle noise in the final geometry approximation. Segmentation is carried out on the reference image based on a predefined thresholding on the color similarity of neighboring pixels or by more sophisticated algorithms (e.g., the nonparametric estimator in [CM02]). Then, depth estimation is performed for the segments rather than on single pixels or pixel windows. Segmentation specialized for image-based rendering applications has been studied in [ZK07]. Geometry reconstruction for image-based rendering does not focus on true geometry, but rather on an appealing appearance of a virtual view generated with the scene geometry. Oversegmentation is often used to further reduce the noise in the produced depth maps.

Also combinations of the above techniques can be found in the literature. A combined silhouette and window-based approach is presented in, e.g., [IS03] or [HS04]. Also depth information retrieved from laser-scanning hardware and geometry reconstructed from images can be combined. Such sensor data fusion techniques to obtain more reliable output have been reported, e.g., in [BAT03, ZP04b, BCS05].

2.7 Compression of image-based scene representations

Beside the various advantages of image-based rendering approaches like photorealism, simple acquisition setup, and scene independent rendering complexity, the downside is the large amount of reference image data that has to be acquired, processed, stored, and possibly transmitted. Raw data sets representing real scenes can be in the tens of Gigabytes to provide acceptable quality for walkthrough applications [AFYC03, ESK05, SCK07]. Downloading these amounts of data is infeasible even over fast network connections. In general, there are three approaches to reduce the overall data size. The first one is to reduce the dimensionality of the representation and to restrict the viewing space. Instead of sampling the full 7D plenoptic function and providing full degree of freedom for navigation in space and time, a representation that meets the requirements of the application, e.g., looking at an object from outside a convex hull might be sufficient (compare to Section 2.1 on page 7). Then, a static scene captured as a light field fits the requirements. The second approach for reducing the data size is to sample the scene representation critically in the sense that no aliasing artifacts are visible in the rendered view as discussed in Section 2.3 on page 11. This also covers the possibility to include scene geometry into the representation which in turn has to be stored to perform view reconstruction from non-uniform samples as discussed for the adaptive sampling approaches in Section 2.3.2 on page 15. Finally, even for undersampled representations there is usually still a lot of redundancy in the captured images that can be removed. Beside the high correlation between neighboring pixels in perspective images, the similarity between neighboring images can be exploited for compression. Further, if allowing deviations between the reconstructed images and the original images, then, lossy compression can be performed to further reduce the information that actually has to be stored. Lossy compression not only removes redundant information in the images, also details considered to be irrelevant for the human visual perception (like high frequency components) are removed. As lossless compression only achieves very small compression ratios (about 2:1) the compression schemes discussed in the following sections solely employ lossy compression.

2.7.1 Random access to samples of the plenoptic function

As mentioned in former sections, virtual view generation consists of the collection or reconstruction of samples of the plenoptic function with respect to the viewpoint and viewing direction of a virtual camera. Usually, samples are captured as whole images. For structured representations the location of a single pixel can be obtained from a simple predefined mapping from a ray parameterization to a storage location. For unstructured representations a search among the camera view points and viewing directions has to be performed to identify the nearest rays from which the desired sample can be interpolated from. Compression schemes greatly reduce the storage size, but usually destroy these structures to exploit intra and inter-image correlation efficiently, e.g., by prediction. To ensure real-time rendering, a fast and random access to image data has to be provided. Efficient compression algorithms have to take this random access requirement into account. Further, compression schemes usually distribute the computational complexity asymmetrically between the encoder and the decoder. In the subsequent analysis it is assumed that encoding can be arbitrarily complex and done offline while the decoding should be as fast as possible.

2.7.2 Intra-image compression

Vector quantization [Gra84, GG91] was one of the first approaches for compression of light fields [LH96]. Strictly speaking, in this scheme not only intra-image redundancy is exploited as every codeword maps to a vector containing intensity values of 48 pixels spanning the 4D parameterization of a light field. This vector has a relatively small support in all four spatial directions, thus, this technique is regarded as intra-image compression. Also for concentric mosaics this approach was adapted [SH99]. Using fixed length codes and simple lookup operations during decoding, this technique provides fast rendering from a compressed scene representation. The coding efficiency of about 10:1 to 20:1 is rather low at acceptable quality. This can be compensated to some degree by entropy coding using Huffman coding [Huf52] or arithmetic coding [WNC87] in conjunction with codebook based general purpose compression schemes like the Lempel-Ziv algorithm [ZL77]. With this further compression of the vector quantized representation, ratios of about 100:1 can be achieved, but, unfortunately, the well structured representation is destroyed.

Transform coding techniques achieve a better compression efficiency than vector quantization at the cost of a higher decoding complexity. Still image compression schemes like JPEG [IJ92] based on a Discrete Cosine Transform (DCT) [CF77], quantization and Huffman coding [Huf52] are adopted for light field compression where a compression ratio of around 30:1 is reported [MRP98] for independent encoding of light field images.

2.7.3 Inter-image compression

The main contributions to the differences between nearby captured images are due to parallax, occlusions and disocclusions, non-Lambertian effects like specularities, and sensor noise. **Prediction-based techniques** exploit inter-frame correlation by parallax compensation where a considerable gain in compression efficiency can be achieved compared to single image encoding as has been shown in theory for common hybrid video encoding concepts [Gir87, Gir93, Gir00] and also for image-based scene representations [TG00, ZC04] with compression ratios of about 500:1 and more at acceptable quality for object centered scenes. Two principal ways of compensating parallax can be found in the literature. The first one is adopted from video compression schemes and referred to as motion (or disparity) compensating prediction where inter-image motion information is provided with the scene representation. The second way directly uses a geometric model of the scene, if provided, to calculate the inter-image motion information used for prediction.

Without a geometric model at hand, parallax can be handled using motion compensated prediction. Input images are partitioned into pixel blocks. While some blocks are independently encoded, others are predicted from one or more blocks in a neighboring image. The reference blocks are simply shifted to their new position in a predicted frame to form the prediction signal. The shifts are represented by motion vectors which form the motion field of a frame. The remaining difference between the predicted and the original image is quantized and encoded or left uncoded if the error is small enough. If the error is too big, the block is encoded independently. The decision how a block is encoded usually is made by means of rate-distortion optimization [Sha48, Sha59, Ber71] using a Lagrangian cost function as in [SW98] for common video. Occlusions, disocclusions, non-Lambertian effects, sensor noise, and disparity compensation inaccuracies are compensated by the residual error that

has to be stored along with the motion vector per block. This technique is used in modern standard video compression schemes like those from the H.26x and the MPEG family [ITJ94, ITU00, Joi03] with modification like variable block size, adaptive quantization, various entropy coding schemes, etc.

Generally, input images for image-based scene representations are often captured as video sequences or can be interpreted as such a sequence. This suggests compression using standard techniques, but, as pointed out in [LH96] it should be avoided to decode from such a compressed representation due to the random access requirement. To access a single block would force to decode all blocks that this single block is predicted from including further dependencies. Nevertheless, with caching of decoded pixel data and other modifications, common video coding algorithms have been successfully adopted for image-based scene representations. E.g., one such modification is that for static scenes a one dimensional disparity displacement is estimated per image block rather than a two dimensional motion vector. This is reasonable since for static scenes and due to the epipolar geometry [LH81] only one dimensional parallax is observed. A scheme based on disparity compensation and fixed length encoding for tree structured vector quantization of the residual error has been investigated in [TG03] with compression ratios of about 200:1 for object centered scenes. In [ZL00, SNC05] pointers to storage locations of whole compressed image parts like large blocks or columns are used to provide random access to the reference image data. Though the compression ratio is lower than for common video compression (about 70:1 for view centered scenes and about 200:1 for object centered scenes) when using prestored pointers, compared to conventional still image coding, the compression is still more efficient. Providing access to whole images rather than to small parts of an input image, in [MG00a] two coders have been presented that use a fixed prediction structure among images of a light field. The first coder employs standard video coding concepts like multiple reference prediction and encoding while the second coder uses a hierarchical structure and more sophisticated disparity estimation which is similar to view dependent geometry reconstruction on a block basis. Both coders achieve high compression ratios of about 1000:1 for an object centered real world scene. In [LWLZ02b] compression for concentric mosaics based on a wavelet decomposition has been presented using a very high level disparity compensation scheme and providing scalability. Random access is still very complicated in this scheme.

When a geometric model is provided with the scene representation, the disparity of pixels or pixel blocks can be directly computed from this model and the camera calibration as in [MEG00a] where compression ratios of about 1000:1 and more on object centered scenes are reached. In [RFG01] a multi-hypothesis prediction framework is used that allows simultaneous prediction from more than one reference frame to improve the disparity compensation accuracy yielding slightly better performance compared to prediction from a single reference as has been shown in theory in [Gir00]. Warping-based compression consists of view dependent texture mapping steps during encoding. In [MRG03] a coding scheme using multiple view dependent textures obtained by projecting light field images onto a geometric model is introduced. This system provides quality and resolution scalability based on a 4D wavelet decomposition, enabling progressive decoding. The compression efficiency increases slightly compared to schemes which do not use geometry, especially for very low bitrates. A drawback with this system is that the input images have to be resampled which bears a slight loss in overall efficiency [CZRG06].

In [MRP98] a surface light field is constructed by resampling of the input images using

scene geometry. This surface light field can be compressed using transform coding or factorization techniques [WAA+00, CBCG02]. Disparity compensated lifting, a scheme adopted from motion compensated lifting used in video coding [ST01], is performed in [CZRG06] not suffering from quality losses during resampling as other compression schemes based on image warping. Additionally, a special adaptation to object centered scenes is made by introducing a shape adaptive compression which takes object boundaries into account and leaves background areas uncoded.

The geometric model or disparity information has to be compressed and stored along with the image information. As shown in [GEM+99, RG02] the bit allocation between geometry and image data has to be considered to achieve optimal compression. Generally, more accurate geometry or disparity information leads to better prediction accuracy but increases the overhead for storing the geometry and disparity as side information.

Different geometry representations can be found for image-based scene representations. Disparity maps are often subsampled to block resolution and used for both the compression without and with geometry, and correspond to a motion vector field in conventional video coding. Per pixel depth images are used, e.g., for layered representations. Explicit geometry information can be provided as a mesh or voxel representation.

Per block disparity maps for image-based rendering are usually fixed length encoded or entropy coded just like for video coding (e.g., [MG00a, SNC05, ZL05]). For arbitrary depth images, encoding has been studied in [KCTS01] using the JPEG2000 [TM02] still image compression standard. There, the dynamic range of the depth images is compressed in a preprocessing step which is based on the fact that view reconstruction is more sensitive to depth inaccuracies for objects nearer to the virtual camera similar to the findings in [CCST00]. A progressive representation of meshes that can be used to trade-off geometry accuracy against storage bit rate has been proposed, e.g., in [Hop96, MG99]. For layered depth images voxel-based geometry is compressed in, e.g., [LIZ+04] using a binary volumetric octree representation [Zhi01]. Also for layered depth images in [DL03] a wavelet based scheme is used for compression.

However, heuristics aiming for low complexity decoding and high compression ratios as well as rate-distortion optimization [Sha48, Sha59, Ber71] are at the core of most of the compression techniques. But, interactive remote walkthrough systems do not only require high compression ratios while preserving a high quality reconstruction and random access to arbitrary images, they also have to take channel characteristics into account as discussed in the next section.

2.8 Streaming of image-based scene representations

Interactive streaming from a server to a client using compressed image-based scene representations is highly related to the compression approaches discussed in the former section, mainly due to the random access problem. When the reference image data has been compressed dependently, e.g., using disparity compensated prediction, generally more pixel data than actually needed for rendering has to be transmitted, decompressed, and possibly discarded if no caching is performed at the client side. Even with caching, typically not all the data is needed during a remote session. This motivates for efficient compression and the possibility to transmit only a small arbitrary part of the whole compressed scene representa-

tion. Packet losses and the transmission delay also have to be taken into account to provide a realistic walkthrough experience. While compression mainly focuses on coding efficiency for local storage, in a streaming scenario, the rate for the transmission of compressed image data from a server to a client is considered. But, unlike for video streaming, the storage rate and the transmission data rate can differ significantly. Nevertheless, streaming of image-based scene representations is also highly related to video streaming.

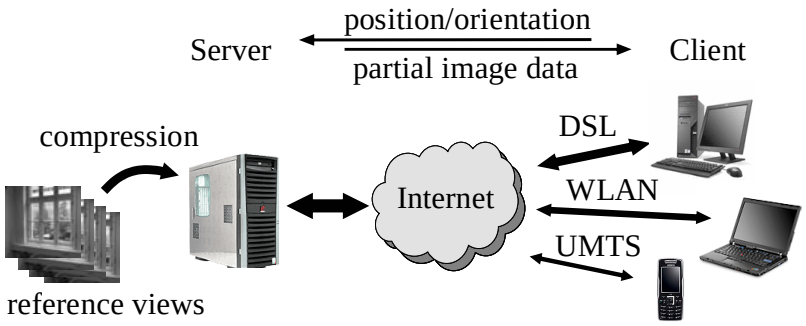

Figure 2.5: *An image-based scene representation is acquired, compressed online or offline, and distributed by a server. Clients with different network access and computational resources are used to remotely navigate in the compressed photorealistic virtual scene.*

A typical interactive streaming system for remote navigation in image-based scene representations over the Internet is depicted in Figure 2.5. During a remote session a user specifies a virtual view position and viewing direction which is signaled to a server. The server responds with the data that can be used to approximate the virtual view. The decision which parts of the compressed bitstream have to be sent is made with respect to the network state like channel throughput, packet loss rate, or delay and with respect to the expected distortion that is introduced when choosing only a subset of all possible data packets that would improve the visual appearance of the desired virtual view. This mechanism is related to bit allocation problems. More transmission data rate is assigned to data that decreases the distortion in the virtual view. I. e., no bits are assigned for the transmission of blocks that are not needed for rendering and the available bit rate is distributed among the image data that is needed. To provide the flexibility for reassembling the compressed bitstream, again, random access functionality has to be considered during encoding.

For still image compression JPEG2000 [TM02] offers the opportunity to rearrange the bitstream of a compressed image to allow dedicated streaming of arbitrary image parts in different resolutions and quality. JPEG2000 is used in some streaming systems that are related to image-based rendering (e.g., [KCTS01] for depth image compression and [ZBTC06] for depth image and texture compression). In [PS01] a proprietary system is described also using a 4D wavelet decomposition on the 4D light field light ray parameterization, thus providing a scalable bitstream. In [CG04] the transmission of arbitrary parts of the data is rate-distortion optimized with respect to the navigation decisions taken by the user and the client's buffer state. The above schemes do not fully exploit inter-image redundancy and therefore do not show optimal performance but provide at least potential streaming capabilities.

Compression and potential streaming for light fields based on distributed encoding has

been investigated in [ZAG03, ARG04]. In these schemes a small representative subset of input images is encoded independently. But, instead of transmitting a residual error for the remaining predicted images, channel coding is applied to allow the decoder to reconstruct the desired pixel data from a prediction formed from any of the neighboring images (including the possibility not to use any of them), thus, providing random access. This scheme has been developed for compression, but, streaming is also possible, though bit allocation during online operation is not considered. In [JSA03] residual error images that have been encoded for all possible prediction directions for all images in a 2D light field setup are stored. At run time, and depending on the user trajectory, the appropriate disparity information and residual error image are transmitted. Additionally, a channel code is used to ensure identical reconstruction of predicted frames independent from the prediction direction. However, the reconstruction process is very complex, though random access as simple as for independent encoding is provided.

Streaming of concentric mosaics based on common video compression concepts, like those used in MPEG video coding (e.g., [IJ94, ITU00]), is introduced in [ZL01, ZL05]. In these systems, independently encoded frames are placed evenly spaced within the captured reference image sequence. Remaining frames and block columns are encoded using disparity compensated prediction from the nearest independently encoded frame while the prediction error is encoded using Huffman coding [Huf52]. An index table is provided with the data set to ensure that the bitstream of pixel columns can be accessed immediately. The major drawback of this system is the high adaptation to concentric mosaic data as well as the fixed prediction structure which does not allow for rate-distortion optimization for compression. Also using disparity compensated prediction, but encoding the residual error using vector quantization encoded with fixed length codes is proposed in [TG03]. Again, streaming is possible in principle, but, rate-distortion optimization is not performed.

Based on the hierarchical compression schemes for light fields in [MEG00a, MRG03], in [RKG03] a rate distortion optimized streaming approach is presented. Video streaming concepts from [CS02, CM06] are adopted to the light field representation. Computationally complex online scheduling is performed to optimize data packetization with respect to the network state. A server or receiver driven user trajectory prediction [RG05] as well as view dependent distortion is taken into account [RKG07], achieving significant gains over a heuristic approach. Streaming using multiple representations has been proposed in [RG04b, RG04a]. Adopting the video coding concepts of SP and SI frames [KK03], the packetization of dependently encoded image data is optimized. SP and SI encoding ensures a prediction direction independent random access to efficiently encoded images. The major drawback is the significantly lower compression efficiency due to the need to store multiple representations.

2.9 Summary

The former sections reviewed a variety of scene representations which belong to both image-based rendering and geometry-based approaches. In fact a clear separation is not possible in many cases and a convergence of computer vision and computer graphics can be observed [Len98]. A general classification of scene representations can be made with respect to the regularity of the intensity sample positions. Structured representations as used for streaming in Chapters 4, 5, and 6, are densely and regularly captured on lines or grids. Usually,

special hardware is used for acquisition and a high reconstruction quality is obtained when sampling and rendering is done according to the theoretical requirements in Section 2.3 on page 11.

Unstructured representations, as used for sparse sampling in Chapter 3, do not constrain the placement of the capturing cameras to be regular. The advantage of unstructured representations is the simple acquisition procedure when a vehicle or a hand-held device is used, but calibration and pose estimation can be challenging. The reconstruction quality is usually moderate because registration errors or mismatches during geometry retrieval result in noticeable and annoying artifacts for sparsely sampled scenes. Robustness is therefore one of the most desirable properties of an acquisition and rendering system. 3D reconstruction from images has been studied quite a long time and research is still ongoing. Even the most recent methods do not achieve satisfactory results in many cases (compare to [SS07]). Existing compression schemes for image-based rendering feature either low compression efficiency at fast random access or high compression performance at a very high decoding complexity without the possibility to trade-off these measures. Streaming has been performed mostly on a best effort basis rather than adaptively with respect to scenario specific properties like channel throughput and the computational capabilities of the client devices. One exception is the work in [RKG03, RG04a] where the transmission data rate vs. quality trade-off is optimized.

Acquisition and rendering with a multi-sensor platform

In [ZC03a] goals for the acquisition of image-based scene representations have been proposed (among others):

- Ease of setup, control and calibration
- Low storage and short capturing time
- Robustness

Similarly, in the original work on unstructured Lumigraph rendering [BBM+01] goals for image-based rendering systems are described (among others):

- Use of geometry information
- Unstructured placement of capturing cameras
- Continuity during motion of the virtual camera
- Resolution sensitivity
- Real-time rendering

With respect to these goals, only a few systems are capable of the acquisition and rendering from a hand-held device for complex scenes. The main research reported has been carried out under lab conditions. Further, in most rendering systems object centered scenes are considered. This fact usually makes pose estimation easier and improves the geometry reconstruction accuracy. As discussed before, hand-held acquisition of view centered representations has been investigated, e.g, in [KPG00] and with a multi-camera system in [ESK05]. While the former system uses a single camera and roughly structured camera movement during acquisition, the latter system is closely related to the multi-sensor system investigated in this thesis. However, the system in [ESK05] reported temporal inconsistent rendering during motion of the virtual camera when using view dependent geometry and texture

mapping whereas the approach using multiple local models produced temporal consistent renderings, but shows holes where depth estimation failed and is not robust to errors in the geometric model.

With the acquisition device and procedure proposed in this thesis, as well as subsequent processing like an improved geometry reconstruction and rendering system, these issues are addressed. This is done by the additional use of a laser line scanner that is calibrated with a three camera system. The multi-sensor setup improves the accuracy of both the image registration and the depth estimation. Remaining outliers are removed to some degree during real-time rendering with the use of local geometry and robust texture blending. This system is presented in Chapter 3.

Efficient streaming of the resulting scene representation based on transmission and decoding of whole images and their depth maps can be performed with the approaches proposed in, e.g., [RKG03, RG04b] and therefore is not considered in this thesis.

Receiver and channel aware compression and interactive streaming of densely sampled scene representations

In addition to the goals defined in the previous section, interactive streaming for remote walkthrough applications should be considered and therefore the following goal for the design of image-based rendering systems is added:

- streaming capabilities with
  - adaptation to the available channel bitrate
  - adaptation of the client computational capabilities
  - progressive transmission and view generation

Most compression and streaming systems for image-based scenes assume object centered representations. Unfortunately, many adaptations made to handle such scenes such as the use of a global geometry model extracted from object silhouettes as well as shape adaptive compression can not be used for view centered scenes in a straight forward manner. Further, in [TG03] typical data access patterns for interactive rendering are investigated and results show that image data needed during interactive streaming differs significantly from the access to whole images as is assumed most often. The systems in [RKG03, RG04b] only allow to place the virtual camera on a hemisphere, though they provide efficient rate-distortion optimized streaming. Another drawback of these systems is that online scheduling has to be performed which is very complex. Additionally, side information like rate-distortion tables might have to be precomputed and stored with the scene representation.

For streaming of densely sampled image-based scene representations there is another relevant work similar to the one proposed in this thesis. In [ZL00] the compression of concentric mosaic representations based on block encoding and image prediction is proposed. The extension to interactive streaming is given in [ZL05]. The complexity for decoding an arbitrary data part is fixed due to the also fixed coding structure. This does not allow to achieve optimal coding efficiency. Further, as already suggested by [TG03] the decoding complexity should be considered during encoding of the scene representation. They suggest - but do not further investigate - to extend a Lagrangian formulation for rate-distortion optimized compression to a rate vs. distortion vs. decoding complexity trade-off which is investigated

for video streaming [vdSA05] in a similar manner:

$$J = D + \lambda \cdot R + \tau C. \tag{2.8}$$

Here, J denotes the cost for encoding the least decodable unit of the input images, D denotes the distortion, R is the (storage) rate, and C denotes the complexity for decoding the image part under consideration with respect to dependencies due to dependent encoding like disparity compensated prediction. λ and τ control the rate vs. distortion vs. complexity trade-off.

So far, no approach is reported in the literature that presents appropriate models for the decoding complexity of compressed image-based scene representations like concentric mosaics and light fields. Further, the incorporation into the optimization procedure during encoding is missing. This thesis closes this gap for the compression and interactive streaming using hybrid video coding concepts. In addition to the incorporation of the decoding complexity into the conventional rate distortion optimization framework, an expected mean transmission data rate is considered. A theoretical analysis is given in Chapter 4 and practical issues are discussed in Chapter 5. Progressive view generation is investigated in Chapter 6.

3 TRIVIS - Image-based rendering using a hand-held multi-sensor platform

In this chapter an approach for the acquisition, processing, and rendering of real scenes, captured using a precalibrated hand-held multi-sensor device, is investigated. The acquisition device consists of a mobile platform, carrying three video cameras and a laser range finder. The images acquired during random motion of the device are used for pose estimation, 3D scene reconstruction supported by the range finder, and for rendering jointly using geometry-based and image-based techniques.

In Section 3.1 an overview of the acquisition, modeling, and rendering approach is given. Section 3.2 is dedicated to the joint calibration of the device sensors, including estimation of the intrinsic camera parameters and their relative pose. In Section 3.3, for completeness, the color calibration procedure used to support proper operation of the techniques described in later sections is discussed. Section 3.4 introduces the pose estimation procedure on which the geometry reconstruction algorithm discussed in Section 3.5 relies on. A rendering method that uses the resulting scene description is evaluated in Section 3.6. Finally, in Sections 3.7 and 3.8 the insights presented in this chapter are discussed and a summary of this chapter is given.

3.1 Scene acquisition and rendering with TRIVIS - Overview

Hand-held acquisition of image data for rendering of image-based and geometry-based scene representations of static real scenes is mostly done using video cameras that usually capture 20-30 frames per second. If the acquisition is done without accurate positioning aided by, e.g., a mobile vehicle or robot arm, images are densely sampled only along the motion trajectory. To sample a scene, e.g., approximately on a 2D grid as in, e.g., [HPDvG99], the camera has to be moved in front of an object in serpentines. This bears the risk of oversampling certain scene regions while others are undersampled. With multi-camera systems, for every acquired image set, multiple viewpoints are covered for every capture position. Existing multi-camera systems for image-based rendering (e.g., [ESK05]) have proven to make image acquisition convenient especially for outdoor scenes. Beside the well defined maximum spacing between two nearest neighboring views, the acquisition process becomes more efficient when multiple images are captured per single shot. While in [ESK05] the cameras are arranged on a line, the multi-sensor platform "TRIVIS" (trifocal vision) as shown in Figure 3.1 carries three video cameras arranged on a triangle. With every exposure, assuming advantageous scene properties, the images captured from three different viewpoints are sufficient to reconstruct parts of the scene and thus to render arbitrary viewpoints, e.g., on the 2D area the triangle spans.

Additionally, TRIVIS makes use of a laser range finder that is intended to support 3D scene reconstruction. The laser scanner can not be used for pose estimation as it only captures

scene points within a single plane and therefore does not allow for feature tracking across single shots, in general. While elegant solutions for self-calibration and pose estimation exist for single-camera and multi-camera systems (see Section 2.5), the cameras and the laser scanner of TRIVIS have to be calibrated before acquisition.

The precalibration process of TRIVIS consists of a joint approach including both, data from the laser scanner and the three cameras simultaneously, where automatically detected point correspondences between the cameras obtained from images of a laser pointer are used. The camera to laser correspondences needed for calibration are obtained manually. After calibration, all intrinsic parameters as well as the relative position of the cameras and the scanner are known.

The calibrated device is used to acquire images of a real scene by moving the device in front of the scene. The device pose is estimated for every single shot by matching feature points in 3D rather than in 2D as done in common structure from motion techniques.

With the recovered structure and motion, dense depth estimation is performed on a subset of all captured images using globally optimized intensity matching. The selection of the image subset used for the estimation of the depth map of every input image is done based on locality and scene visibility. The obtained depth maps are simplified to a coarser 3D mesh representation using a mesh simplification technique.

The view dependent 3D mesh and texture is then fed into a rendering engine that performs view warping and view interpolation on a small subset of the input images based on the users viewing position to generate a virtual view in real-time. Again, the subset of used images is selected based on locality and scene visibility. To suppress artifacts due to unreliable depth values obtained during the 3D scene reconstruction process, an on the fly outlier detection and removal is performed.

Figure 3.1: *TRIVIS - A three-camera system attached to a laser scanner*

3.2 Joint laser-camera calibration

In this section the geometric calibration of TRIVIS is described. While multi-camera calibration techniques (compare to 2.5.1 on page 20) only calibrate the system up to scale when prior knowledge about the captured scene or the used calibration pattern is unknown, the laser scanner supports a metric reconstruction of the intrinsic and extrinsic parameters of TRIVIS including the scale. A further advantage of the method described in this section is that the almost degenerate configuration of the cameras can be recovered which is not always possible with other calibration techniques.

3.2.1 Device setup

TRIVIS consists of three IDS [IDS07] USB high resolution video cameras (1280×1024 Bayer-Pattern). The field of view using high accuracy optics is approximately 30 to 40 degrees. Images are synchronously captured for all three cameras at a frequency of 2Hz. The SICK-LMS295 scanner [SIC07] is a line laser scanner designed for industrial applications. In the considered configuration, the scanner fires laser beams spaced quarter a degree over a field of view of 100° and detects their recurrence. Such 100° line scans are captured at 20Hz. The sensor setup is illustrated in Figure 3.1 and 3.2. The video cameras are arranged on a triangle with a side length of approximately 0.25 meters. The centers of projection are lying approximately on the X-Y plane. The cameras face in $-Z$ direction and the projection axes are slightly converging (meeting approximately 2 meters in front of the device).

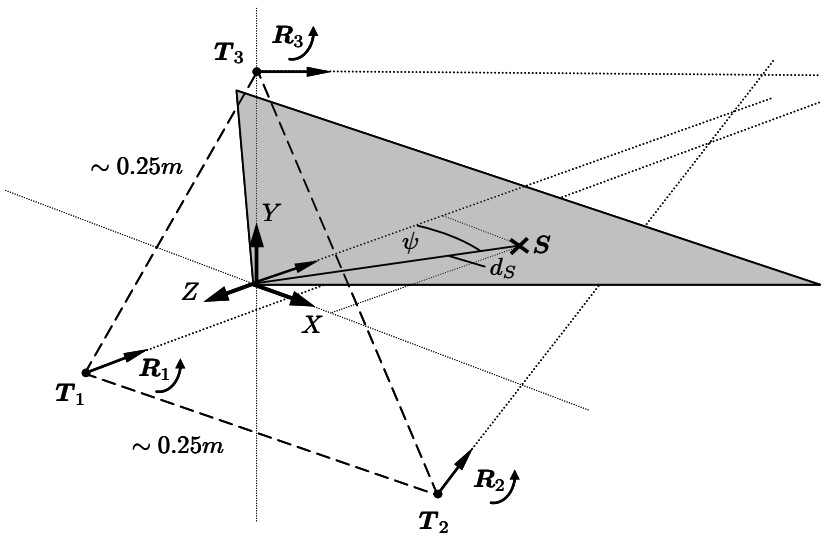

Figure 3.2: *TRIVIS: Capture geometry seen from the upper right behind the device. The device coordinate system coincides with the laser scanner coordinate system. The cameras are arranged on a triangle with approximately 0.25 meters side length on the X-Y plane. The sensors face in $-Z$ direction and their projection axes converge approximately 2 meters in front of the device.*

3.2.2 Sensor models

Camera model The cameras are modeled as pinhole cameras (e.g., [Pol00, HZ04]). These perspective cameras map a world point denoted as 4-vector $\boldsymbol{X}$ in homogeneous coordinates to homogeneous image coordinates $\boldsymbol{x}$ using the projection defined by the 3×4 matrix $\boldsymbol{P}_i$ of camera i:

$$\boldsymbol{x} \sim \boldsymbol{P}_i \cdot \boldsymbol{X}. \tag{3.1}$$

This mapping is determined up to scale as expressed by $\sim$. For a Euclidean world frame, $\boldsymbol{P}_i$ can be factorized as

$$\boldsymbol{P_i} = \boldsymbol{K}_i \left(\boldsymbol{R}_i^T \middle| - \boldsymbol{R}_i^T \boldsymbol{T}_i \right) \tag{3.2}$$

where the 3-vector $\boldsymbol{T}_i$ and the 3×3 matrix $\boldsymbol{R}_i$ are the camera translation and rotation, respectively. The 3×3 intrinsic calibration matrix $\boldsymbol{K}_i$ of camera i is defined as

$$\boldsymbol{K_i} = \begin{pmatrix} f_{ix} & s_i & c_{ix} \\ 0 & f_{iy} & c_{iy} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

where the focal lengths f_{ix} and $f_{iy} = \alpha_i \cdot f_{ix}$ (α_i denotes the aspect ratio) and the principal point $(c_{ix}, c_{iy}, 1)^T$ are measured in pixels (px), and s_i denotes the pixel skew. Non-linear radial distortion can be taken into account by introducing a mapping $\boldsymbol{K}_i^{nl}(\boldsymbol{x})$ that depends on the 3D coordinates of a scene point with respect to the camera frame. With the first order radial distortion correction coefficient r_i^{nl} usually accounting for about 90% of the total distortion [MT96], Equation (3.2) is augmented as:

$$\boldsymbol{P_i} = \boldsymbol{K}_i \boldsymbol{K}_i^{nl} \left(\boldsymbol{R}_i^T \middle| - \boldsymbol{R}_i^T \boldsymbol{T}_i \right) \text{ with } \boldsymbol{K}_i^{nl}(\boldsymbol{x}) = (1 + r_i^{nl}(x^2 + y^2)) \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{3.4}$$

Due to the limited field of view of the cameras and the high accuracy optics used, the radial distortion is very small and assumed to vanish ($r_i^{nl} = 0$ in Equation (3.4)), as validated by experiment using conventional single camera calibration software [Bou07, SFA07].

Scanner model The laser scanner is modeled as an affine camera. The raw data consists of indices denoting an angle ψ and distance d_S (measured in millimeters rather than pixels) in the X-Z plane of a coordinate system attached to the laser scanner as shown in Figure 3.2. The X-Z plane is defined as the image plane of the virtual scanner camera with its center of projection at infinity in Y direction. $\boldsymbol{s}$ is a 3-vector in homogeneous coordinates and denotes a point in the scanner image plane and can be computed as

$$\boldsymbol{s} = \begin{pmatrix} d_S \cdot \sin\psi \\ -d_S \cdot \cos\psi \\ 1 \end{pmatrix} = \begin{pmatrix} X \\ Z \\ 1 \end{pmatrix}. \tag{3.5}$$

The projection denoted in Equation (3.1) for world points $\boldsymbol{X}$ in the scanner plane is adopted as:

$$\boldsymbol{s} = \boldsymbol{P}_s \cdot \boldsymbol{X}. \tag{3.6}$$

The orthographic projection matrix $\boldsymbol{P}_S$ modeling the laser scanner is:

$$\boldsymbol{P}_S = \boldsymbol{K}_S \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_S^T & -\boldsymbol{R}_S^T \boldsymbol{T}_S \\ \boldsymbol{0}^T & 1 \end{pmatrix} \tag{3.7}$$

with $\boldsymbol{R}_S$ and $\boldsymbol{T}_S$ being the scanner rotation and translation with respect to a world frame. The 3×3 matrix $\boldsymbol{K}_S$ is the intrinsic calibration matrix of the scanner. The laser scanner coordinate system is chosen as the world frame during calibration. Together with the assumption that the laser scanner delivers values not corrupted by any systematic error ($\boldsymbol{K}_S$ is the identity matrix), Equation (3.7) can be simplified to

$$\boldsymbol{P}_S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.8}$$

3.2.3 Calibration pattern

The calibration of TRIVIS is based on point correspondences. These correspondences can be assigned to classes of world points that can be differentiated as:

1. points visible in all cameras and by the laser scanner,
2. points only visible in all cameras, but not by the laser scanner,
3. points not visible in all cameras, but at least in one camera and by the laser scanner, and
4. points not visible in all, but two cameras and not visible by the laser scanner.

Most elegantly, such point correspondences are obtained automatically from the captured data itself. This requires sufficiently overlapping sensor images which is provided by the video cameras, but not between the cameras and the line laser scanner. Instead, a virtual calibration pattern is used. The virtual calibration object is constructed by waving a laser pointer in a darkened environment. The laser spot is detected in synchronously captured images. Such correspondences can easily be established and located very precisely, one for every captured image triplet. This procedure can be automated [SMP05] and points belonging to the second class of correspondences are obtained. Additionally, if the laser pointer crosses the scanner plane, a laser scan is triggered manually to acquire the images of a world point in all four sensors, which produces correspondence of the first class. Generally, correspondences of the first and third class relate to coplanar world points. The acquisition of real world data for calibration is discussed in detail in Appendix A.2 on page 160.

3.2.4 Initial solution

The scanner model is based on an affine camera model, and thus the overall device calibration problem becomes a multi-camera calibration problem where one of four cameras is affine. As the scanner image is chosen to coincide with the device coordinate frame, five intrinsic and six extrinsic camera parameters have to be determined for each video camera summing up to 33 parameters to recover. The extrinsic camera parameters are hereafter called “device extrinsics” as these extrinsic parameters are determined relative to the device

frame. An initial solution is obtained by using plane induced homographies that map points visible to the laser scanner into the image planes of the cameras. From these homographies a partial reconstruction of the camera projection matrices can be obtained. The missing information is then filled in by a common multi-camera self-calibration approach. Finally, the initial estimate is refined by bundle adjustment.

Plane induced homographies The coplanarity of points lying in the scanner plane is exploited by estimating homographies between the scanner image and the camera images. A point $\boldsymbol{s}$ in the scanner image is projected into the camera image planes via: $\hat{\boldsymbol{x}}_1 = \boldsymbol{H}_{s1}\boldsymbol{s}$, $\hat{\boldsymbol{x}}_2 = \boldsymbol{H}_{s2}\boldsymbol{s}$, $\hat{\boldsymbol{x}}_3 = \boldsymbol{H}_{s3}\boldsymbol{s}$. Where the 3×3 matrix $\boldsymbol{H}_{si}$ describes the mapping from the scanner plane to the image plane of camera i. There is a second way of mapping a point $\boldsymbol{s}$ onto the image planes of the cameras. A world point $\boldsymbol{X}$ corresponding to $\boldsymbol{s}$, obtained from Equation (3.6), can be mapped by the camera projection matrix $\boldsymbol{P}_i$:

$$\hat{\boldsymbol{x}}_i = \boldsymbol{H}_{si}\boldsymbol{s} = \boldsymbol{H}_{si}\begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} \sim \boldsymbol{P}_i\boldsymbol{X} = \boldsymbol{P}_i\left(\boldsymbol{P}_S^{+}\boldsymbol{s}\right) = \boldsymbol{P}_i\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\boldsymbol{s} \tag{3.9}$$

Here $^{+}$ indicates the pseudoinverse of a matrix. Equation (3.9) expresses that with their corresponding homographies $\boldsymbol{H}_{si}$ known, the projection matrices of the cameras $\boldsymbol{P}_i$ are also known (up to scale), except for their second column (second row of $\boldsymbol{P}_S^{+}$ is all zero). This is not surprising as this column weights the Y coordinate of a world point - which is not captured by the scanner. It follows that

$$\boldsymbol{H}_{si} \sim \begin{pmatrix} \boldsymbol{p}_i^1 & \boldsymbol{p}_i^3 & \boldsymbol{p}_i^4 \end{pmatrix} \tag{3.10}$$

Where $\boldsymbol{p}_i^k$ denotes the kth column of the matrix $\boldsymbol{P}_i$.

Computation of the homographies A pair of corresponding point measurements $\boldsymbol{s}$ and $\boldsymbol{x}_i$ in the scanner image and a camera image i, respectively, are related via $\boldsymbol{x}_i \sim \boldsymbol{H}_{si}\boldsymbol{s}$. Assuming $\boldsymbol{H}_{si}$ to be a projective transformation, it has 8 degrees of freedom [HZ04]. With 4 point correspondences between each camera image and the laser scanner, this mapping can be determined. In the presence of noise and outliers, more points can be used in conjunction with a robust outlier detection and removal algorithm (e.g., RANSAC [FB81]) to obtain a least squares solution [HZ04].

The obtained linear solution of the homographies is quite noise sensitive, and, as during computation of the linear solution a matrix norm is minimized, the considered measure is not the distance between a point measurement and the projection of its correspondence, as would be desirable [Sch07]. To overcome this drawback, the linear solution is refined by a joint maximum likelihood estimation of the scanner-camera homographies. The following expression is minimized:

$$\sum_{l=1}^{n_c}\left(\frac{\left\|\boldsymbol{s}^l - \hat{\boldsymbol{s}}^l\right\|^2}{\sigma_s^2} + \frac{\sum_{i=1}^{3}\left\|\boldsymbol{x}_i^l - \hat{\boldsymbol{H}}_{si}\hat{\boldsymbol{s}}_i^l\right\|^2}{\sigma^2}\right) \tag{3.11}$$

where n_c is the number of point correspondences (of class one in this case, however, correspondences of the third class can also be incorporated with minor changes in the formalism of (3.11)). $\boldsymbol{s}^l$ is the lth measurement in the scanner image while $\boldsymbol{x}_i^l$ is the corresponding lth measurement in the ith camera. σ_s and σ denote the variances of the noise (assumed to be Gaussian) of measurements in the scanner image and camera images, respectively. The norm of a vector in homogeneous coordinates in Expression (3.11) and all subsequent expressions refers to the 2-norm of the corresponding vector in inhomogeneous coordinates. $\hat{\boldsymbol{s}}^l$ as well as $\hat{\boldsymbol{H}}_{si}$ are estimated by minimizing (3.11) using the Levenberg-Marquardt algorithm [Lev44, Mar63]. For implementation details see [HZ04, Sch07]. After the joint maximum-likelihood estimation of the homographies, the projection matrices $\boldsymbol{P}_i$ are partially determined according to (3.10).

Retrieval of the second columns The image of the absolute conic $\boldsymbol{\omega}$ is an imaginary conic that is used often to retrieve camera parameters (see, e.g., [HZ04] for a detailed analysis). The intrinsic camera matrix $\boldsymbol{K}_i$ can be calculated by Cholesky factorization of the inverse of $\boldsymbol{\omega}$ followed by an RQ-decomposition:

$$\boldsymbol{\omega}_i^{-1} \overset{Chol.}{=} \boldsymbol{L}\boldsymbol{L}^T \overset{RQ}{=} (\boldsymbol{K}_i\boldsymbol{R}_i^T)(\boldsymbol{K}_i\boldsymbol{R}_i^T)^T = \boldsymbol{K}_i\boldsymbol{K}_i^T \tag{3.12}$$

Here, $\boldsymbol{L}$ is a lower triangular matrix, $\boldsymbol{K}_i$ the upper triangular intrinsic calibration matrix of camera i, and $\boldsymbol{R}_i$ is the orthonormal camera rotation matrix. This decomposition is unique if the signs of $\boldsymbol{K}_i$'s diagonal elements are positive.

The partially known projection matrices impose constraints on $\boldsymbol{\omega}$. Using $\boldsymbol{K}_i\boldsymbol{R}_i^T = (\boldsymbol{p}_i^1, *, \boldsymbol{p}_i^3)$ (from Equation (3.2) in the notation of (3.10)) and with the unknown second column denoted by $*$, it follows that

$$\boldsymbol{R}_i\boldsymbol{R}_i^T = \left(\boldsymbol{p}_i^1, *, \boldsymbol{p}_i^3\right)^T \boldsymbol{K}_i^{-T}\boldsymbol{K}_i^{-1} \left(\boldsymbol{p}_i^1, *, \boldsymbol{p}_i^3\right) \tag{3.13}$$

Due to the orthogonality of $\boldsymbol{R}_i$ this implies two constraints on $\boldsymbol{\omega} = \boldsymbol{K}^{-T}\boldsymbol{K}^{-1}$:

$$(\boldsymbol{p}_i^1)^T\boldsymbol{\omega}\boldsymbol{p}_i^1 = (\boldsymbol{p}_i^3)^T\boldsymbol{\omega}\boldsymbol{p}_i^3 \tag{3.14}$$

$$(\boldsymbol{p}_i^1)^T\boldsymbol{\omega}\boldsymbol{p}_i^3 = 0 \tag{3.15}$$

For each of the three cameras with index i there are two constraints and under the assumption that the cameras are identical, the calibration matrices and the images of the absolute conic are also identical: $\boldsymbol{K}_i = \boldsymbol{K}$ and consequently $\boldsymbol{\omega}_i = \boldsymbol{\omega}$. The six constraints on the common image of the absolute conic $\boldsymbol{\omega}$ with five degrees of freedom, lead to an overdetermined set of equations that can be solved for $\boldsymbol{\omega}$ in the least squares sense.

The computation of the image of the absolute conic with full degree of freedom is still noise sensitive. To decompose $\boldsymbol{\omega}^{-1}$ according to (3.12) it must be either strictly positive or negative definite which is not ensured with noisy correspondences. Further, as mentioned in the introduction of this section, the device setup is close to a degenerate case as the camera image planes are perpendicular to the scanner image [SM99, Stu97]. Since the number of simultaneous views is fixed, only the reduction of degrees of freedom for $\boldsymbol{K}_i$ is feasible to make the algorithm more robust. An alternative is to perform the calibration from correspondences of classes two and four using conventional multi-camera calibration software (e.g., [SMP05]), which, in practice has problems with the degenerate configuration of TRIVIS. Additionally, this would not involve the laser scanner in the calibration process.

The most crucial camera parameter is the focal length. All other intrinsic parameters can usually be set to default values according to the hardware (e.g., CCD chip dimensions, resolution, etc.) in order to obtain reasonable results for the initial parameter estimate. The six constraints on the common image of the absolute conic $\boldsymbol{\omega}$ can be used to estimate only the focal length which leads to an overdetermined set of equations that, again, can be solved for $\boldsymbol{\omega}$ in the least squares sense. The initial estimates of the intrinsic calibration matrices $\boldsymbol{K}_i$ are subsequently determined using (3.12).

Applying ${\boldsymbol{K}_i}^{-1}$ to each of the camera matrices $\boldsymbol{P}_i$ yields parts of the transposed camera rotation matrices $\boldsymbol{R}_i^T$.

$$\left(\boldsymbol{r}_i^1, *, \boldsymbol{r}_i^3\right) = \boldsymbol{K}_i^{-1}\left(\boldsymbol{p}_i^1, *, \boldsymbol{p}_i^3\right) \tag{3.16}$$

where $\boldsymbol{r}_i^k$ denotes the kth column of the transposed rotation matrix $\boldsymbol{R}_i^T$. Proper scaling of $\boldsymbol{P}_i$ ensures the orthogonality of $\boldsymbol{r}_i^1$ and $\boldsymbol{r}_i^3$. $\boldsymbol{r}_i^2$ is determined as their cross product. The ambiguity in the sign of $\boldsymbol{r}_i^2$ can be resolved considering the relative position of the cameras to the scanner. The camera centers can be computed from $\boldsymbol{T}_i = -\boldsymbol{R}_i{\boldsymbol{K}_i}^{-1}\boldsymbol{p}_i^4$, and according to the device setup, their Y-component has to be positive if the camera lies above the scan plane, or negative if it lies below.

3.2.5 Refinement

Initialized with the estimate of the camera parameters computed in the former section, a refinement based on bundle adjustment improves the final device parameter estimates.

Up to now, only world points visible in the scanner image have been used (coplanar world points in the scanner plane - correspondence class one and three). To obtain a more precise calibration, points of the second and fourth class of correspondences are also used for global parameter optimization. With the initial parameter estimate and the sensor models, 3D world points can be computed from the N_c correspondences $\boldsymbol{x}_i^l$ in camera i by triangulation (see, e.g., [HZ04]), where $l \in [1, N_c]$. The retrieved point cloud also serves as an initial estimate for the global structure and motion refinement. The entire setup involving N_c 3D points has $3N_c + 33$ degrees of freedom, 3 for each point and 11 per camera.

The objective is to find the set of these parameters that leads to estimated image points $\hat{\boldsymbol{x}}_i$ in the camera images and $\hat{\boldsymbol{s}}$ in the scanner image as close as possible to the measured points $\boldsymbol{x}_i$ and $\boldsymbol{s}$. The parameters are to be recovered according to their maximum likelihood. This is achieved by minimizing the reprojection error in the sensor images, weighted by the corresponding error variance. This procedure corresponds to the minimization of the squared Mahalanobis distance (see [HZ04] for details) that can be formulated as:

$$\sum_{l=1}^{n_c} \frac{\left\|\hat{\boldsymbol{s}}^l - \boldsymbol{s}^l\right\|^2}{\sigma_s^2} + \sum_{l=1}^{N_c} \frac{\sum_{i=1}^{3}\left\|\hat{\boldsymbol{x}}_i^l - \boldsymbol{x}_i^l\right\|^2}{\sigma^2} \tag{3.17}$$

An important advantage of this objective function is that every sensor class is assigned a weight reciprocal according to its variance, and therefore, more noisy measurements have less influence on the calibration result. Equation (3.17) is minimized using the Levenberg-Marquardt algorithm [Lev44, Mar63]. σ and σ_s need not to be known, instead, it is sufficient to specify an appropriate ratio σ_s/σ. The whole calibration process is summarized in Table 3.1.

- Compute scanner plane to camera image homographies $\boldsymbol{H}_{si}$ from 4 or more point correspondences and refine them if necessary.
- Compute the image of the absolute conic $\boldsymbol{\omega}$ and retrieve the intrinsic calibration matrices $\boldsymbol{K}_i$.
- Apply ${\boldsymbol{K}_i}^{-1}$ to the projection matrices $\boldsymbol{P}_i$ obtained from (3.10) and reconstruct the partially known, orthonormal rotation matrices $\boldsymbol{R}_i^T$. Complete the rotation matrices by cross multiplying available columns.
- Refine the initial estimate by bundle adjustment, minimizing the squared Mahalanobis distance of the error vector.

Table 3.1: *Summary of the joint calibration algorithm.*

3.2.6 Accuracy evaluation on synthetic data

In this section the calibration algorithm is evaluated on synthetic data providing ground truth.

Evaluation methodology and measures The geometrical setup of a virtual multi-sensor device is chosen close to the real setup (see Figure 3.2) with additive noise on the intrinsic parameters (Gaussian noise with a standard deviation of 2 pixels for the principal point at (c_{xi},c_{yi})=(640,512), 5 pixels for the focal lengths (f_{xi}=1800 and f_{yi}=1900), and 0.05 for the pixel skew (s_i=0.0)). The camera centers are altered by Gaussian noise with a standard deviation of 2mm. Also, the orientation of the cameras' optical axes are subject to Gaussian noise with a standard deviation of $0.5°$. The obtained setups provide the ground truth parameterization to which estimates are compared during the experiments.

A set of $N_c = 140$ points is generated to serve as the virtual calibration pattern. To mimic real data, a cube of side length 2m that lies 1m in front of the system is defined, where randomly generated world points lie in. $n_c = 40$ of these points produce correspondences of the first class that are visible in all cameras and the laser scanner. 100 of the world points produce correspondences of the second class which are only visible in the cameras. To evaluate the robustness of the calibration algorithm, Gaussian noise is added to the imaged points.

Measurements in the camera images are given in pixels, in the scanner image in millimeters. The noise variance ratio between measurements in the camera and scanner image planes is chosen as: $\sigma_s^2 : \sigma^2 = 9 : 1$. This ratio approximately corresponds to the size of a pixel in the cameras projected onto the scanner plane in a distance of 2m in front of the device. Every

experiment is repeated on 1000 (for the initial estimate) or 100 (for the full calibration) randomly chosen point sets and geometrical setups, and the results are averaged.

The 3D reconstruction error is determined by comparing ground truth world points $\boldsymbol{X}$ with a reconstruction from the point measurements $\hat{\boldsymbol{X}}$ using the estimated parameters and sensor models (the non-linear method for triangulation from multi-view correspondences from [HZ04] is used). The reprojection error is determined by comparing the projections of the ground truth world points, using the ground truth system parameters and models, to the finally obtained projections of the estimated world points and their reprojections with respect to the estimated system parameters.

The algorithm fails to produce reliable results in up to 40% of the pointsets (subsets of the available correspondences) that are fed into the system and having noise levels with a standard deviation greater than 2 pixels. These degenerate configurations are detected from the reprojection error, and a different subset of input points is chosen.

Accuracy evaluation for the initial estimate The homography refinement algorithm described in Section 3.2.4 on page 37 takes about 10 iterations to converge. Figure 3.3 shows the accuracy of the initial parameter estimation compared to ground truth exemplarily for camera 2 and the laser scanner. The results for cameras 1 and 3 are substantially the same as for camera 2. The focal length is recovered quite accurately with an error of about 0.3% for a noise level of 2 pixels. The position of the principal point is commonly considered not to have much impact on the overall results [Tri98, SM99], however, noise below 2 pixels standard deviation leads to a displacement of the principal point of less than 6 pixels which corresponds to 1% deviation with respect to the image resolution. The reconstruction error for the pixel skew is negligible at an order of magnitude of 10^{-4}. The estimated aspect ratio gives a mean error of about 8% for a noise level of 2 pixels and more. This seems to be high, but is still acceptable in practice. The reconstruction error for the device extrinsics and the reconstructed scene points is illustrated in the right of Figure 3.3. The camera positions are recovered with an average error of almost 15mm at a noise level of 2 pixels. For the same noise level the camera orientations differ about 0.3° and the scene points are reconstructed with an accuracy of 25mm.

The reprojection errors are shown in Figure 3.4. The mean squared error as well as the mean error are plotted for all cameras and the laser scanner, respectively. With a mean reprojection error of about 1.5 pixels in the cameras at a noise level of 2 pixels, the algorithm performs comparable to the initial estimate of state of the art multi-camera calibration techniques (e.g., [SMP05, UT03]). The geometrical error on the scanner image plane of around 8 millimeters at a noise level of 2 pixels is near the vendor's specification of an error of 5 millimeters. Overall, the reconstruction and reprojection error are well balanced between the sensors.

Accuracy evaluation after refinement With a refinement using bundle adjustment, the reprojection and reconstruction errors in the camera images and the laser scanner image, respectively, can be further reduced (see Figure 3.5). The model parameters do not change essentially except for a slight increase in the error of the principal point estimate. The algorithm achieves a mean reprojection error of 0.6 pixels in the cameras at a standard deviation of the noise of 2 pixels. In the scanner image, at this noise level, a mean error of 5 millimeters is observed for the synthetic data. Overall, the reconstruction and reprojection error are well

balanced between the sensors. The standard deviation in the mean errors is quite small as indicated by the vertical bars in Figure 3.5. Also the final calibration results are comparable to common multi-camera calibration algorithms.

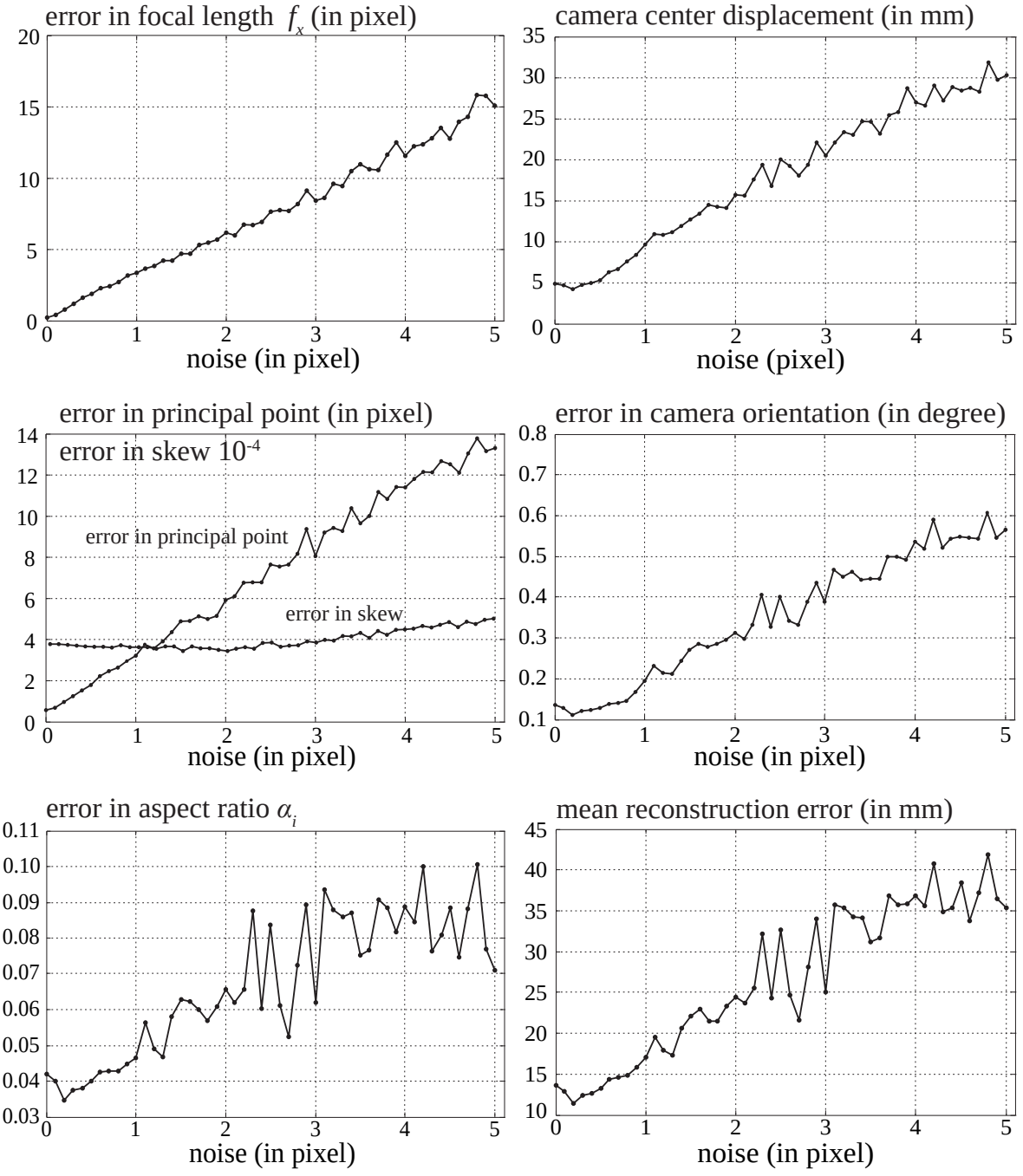

Figure 3.3: *Comparison of errors in the camera parameters for camera 2 (left) and the geometric error of the reconstructed camera position, orientation, and the scene points (right) with respect to the noise level in the measurements.*

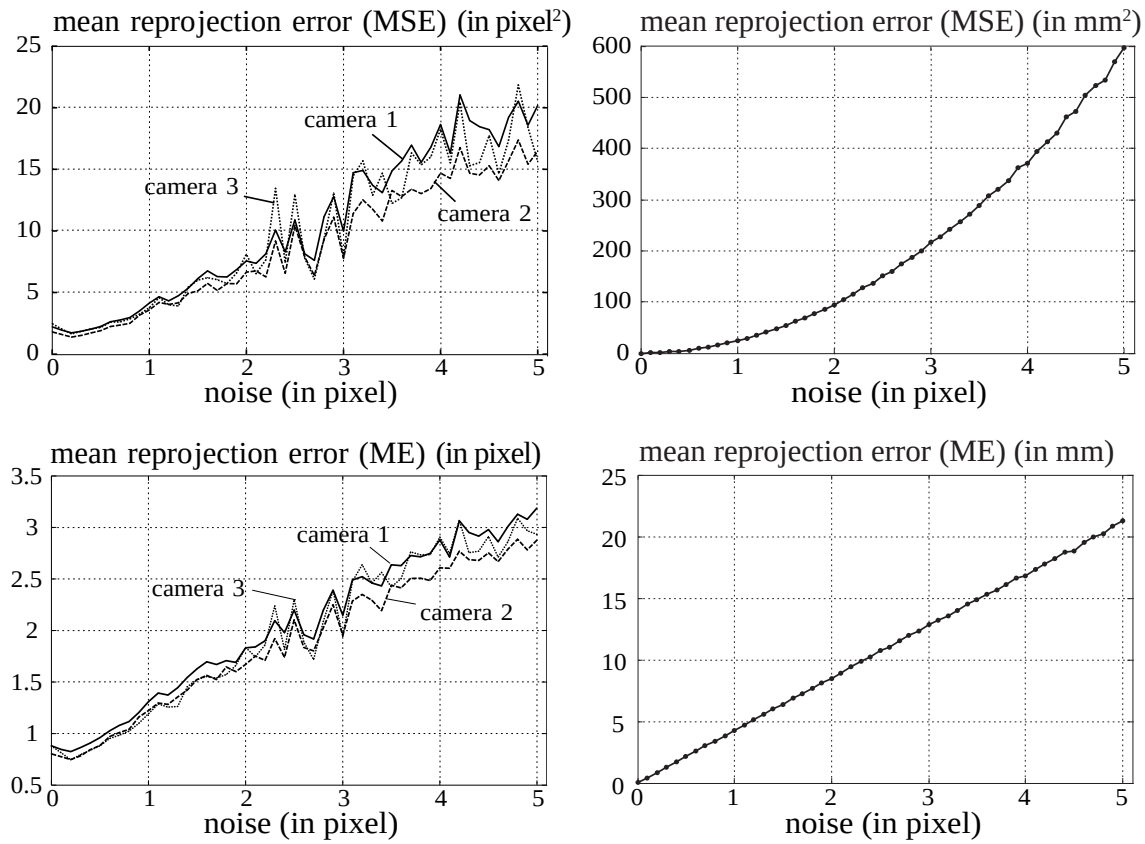

Figure 3.4: *Comparison of the reprojection error in the video cameras (left) and the reprojection error for the laser scanner (right) with respect to the noise level in the measurements.*

3.2.7 Accuracy evaluation on real data

In this section the calibration algorithm is evaluated on real data. The acquisition procedure of the virtual calibration object is described in Appendix A.2 on page 160. The test data set consists of $n_c = 19$ coplanar points on the scanner plane and 20 points freely placed in front of the device, summing up to $N_c = 39$ point correspondences. The images were taken such that all points are visible in all three camera images and, therefore, belong to the first and third class of correspondences.

Points in the scanner image are not easy to detect, even not manually, which leads to an error in the point measurements of about 5 mm. In the camera images, where the laser points can be identified more reliably, the error is assumed to be in the range of 1 to 3 pixels. Based on these values, the error variance ratio is set to $\sigma_s : \sigma = 3 : 1$

The initial parameter estimate is obtained from $n_c = 19$ coplanar points. As outliers among the detected point measures are possible, for this experiment, a robust implementation based on RANSAC is used to determine an appropriate subset of coplanar points to compute the initial estimate from. The maximum likelihood refinement is used where an improvement can be observed in terms of a reduced reprojection error. Figure 3.6 (top) visualizes the reprojection error vectors (as points) for every point measure using the initial estimate of model parameters. Numerical values of the reprojection errors are given in Table 3.2. The

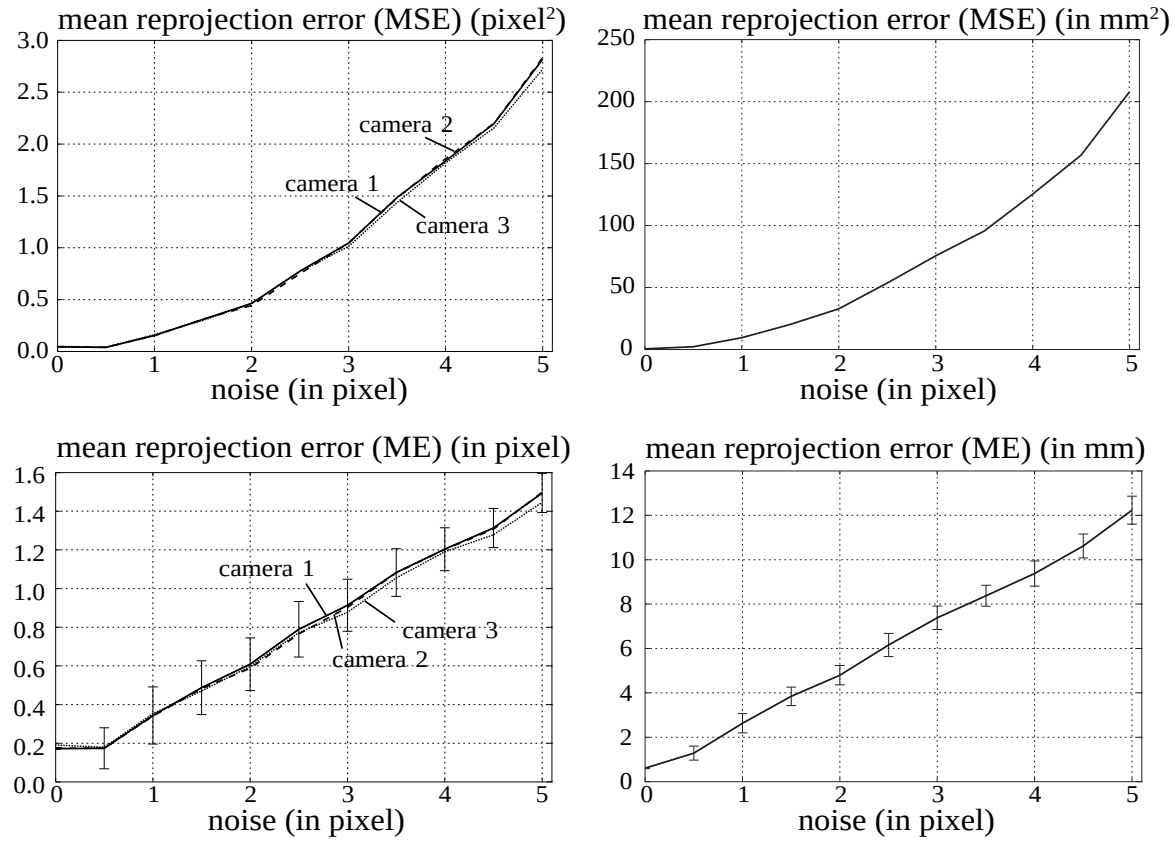

Figure 3.5: *Reprojection error after refinement in the cameras (left) and the scanner image (right) with respect to the noise level in the measurements.*

same scenario is shown in Figure 3.6 (bottom) for calibration including the refinement step. Numerical results with standard deviations are also given in Table 3.2. With a final accuracy of approximately 0.55 pixel for the reprojected world points in the camera images, and with a standard deviation of 0.5 pixel, the results can compete with standard algorithms like [SMP05]. The accuracy of the scanner projections is in the expected range of the vendor's specification of 3.2 mm with a standard deviation of approximately 2.6 mm. Again, the error is well balanced between the sensors.

The unknown variance ratio σ_s^2/σ^2 influences the focus of the bundle adjustment step from the cameras to the scanner and vice versa. Choosing higher ratios would allow us to achieve a lower reprojection error in the images at the cost of a higher reprojection error in the scanner image and vice versa.

3.3 Color calibration

This section briefly introduces the color calibration procedure of the video cameras of TRIVIS. As the device consists of three slightly different cameras, the measured intensity values of the same scene point might vary among the sensors not only due to surface properties like reflectance, but also due to slightly different scene radiance to intensity mappings in the

working chain of the imaging process. Further, as video cameras, in general, can only capture a limited dynamic range, it is desirable to determine the photometric properties of the whole capturing device to build a high dynamic range image from multiple exposures.

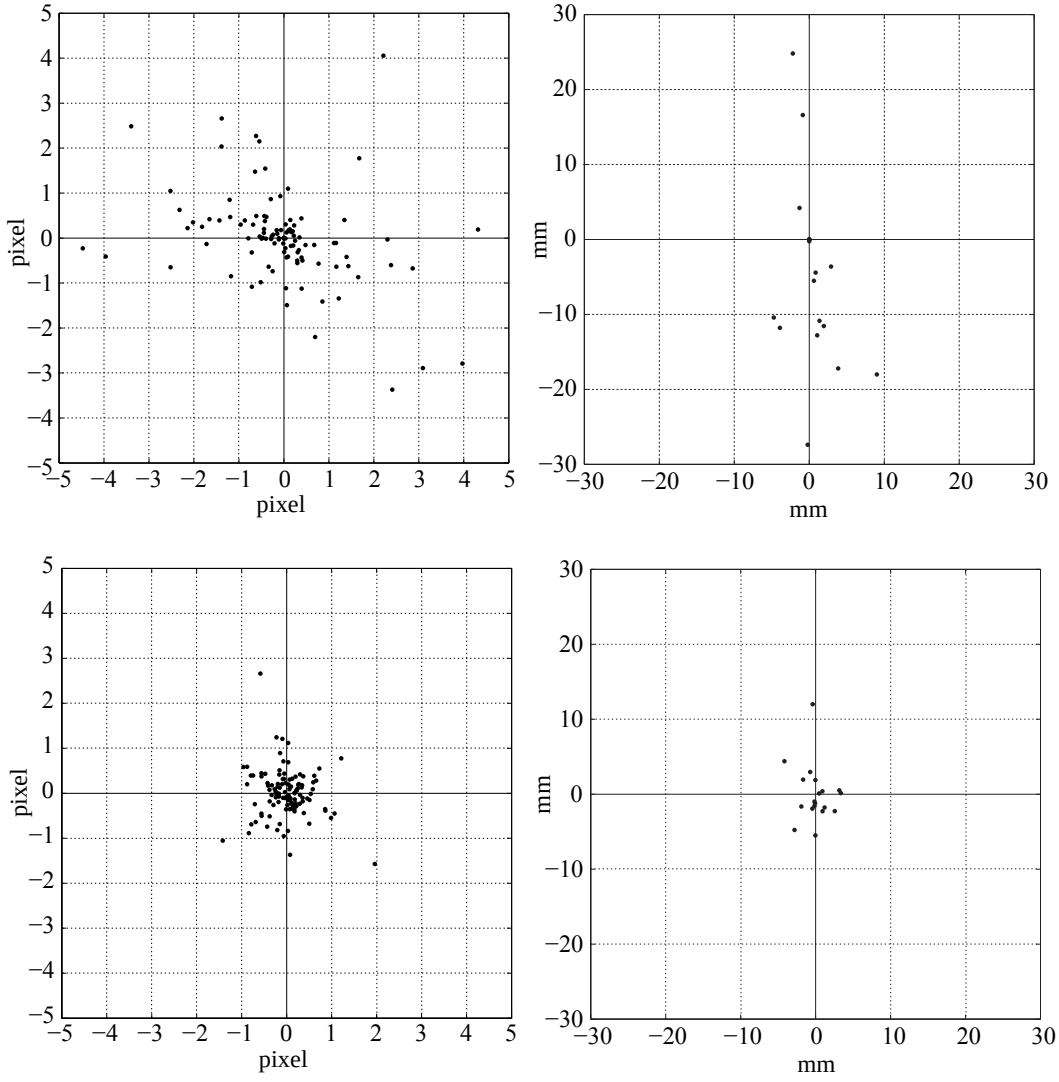

Figure 3.6: *(top) Reprojection error in the cameras and the scanner images after the linear part of the calibration algorithm, visualized as the difference between measured and reprojected points. (bottom) Reprojection error after refinement in the cameras and the scanner images.*

3.3.1 High dynamic range imaging

Especially for outdoor scenes, the brightness of the scene can span several orders of magnitude in the area of interest. Conventional video cameras as those used with TRIVIS can only image a small part of this range. To still be able to capture such scenes, multiple exposures during acquisition could be used. However, a smooth incorporation of multi-exposure images into one common representation is important for subsequent processing steps like

	initial	refined	initial	refined
	MSE		ME	
camera 1	$1.6514\,\text{px}^2$	$0.52517\,\text{px}^2$	$0.87896 \pm 0.94973\,\text{px}$	$0.55649 \pm 0.47027\,\text{px}$
camera 2	$3.2853\,\text{px}^2$	$0.41527\,\text{px}^2$	$1.3015 \pm 1.278\,\text{px}$	$0.53054 \pm 0.37057\,\text{px}$
camera 3	$3.8314\,\text{px}^2$	$0.53668\,\text{px}^2$	$1.3168 \pm 1.4672\,\text{px}$	$0.54812 \pm 0.4924\,\text{px}$
scanner	166.4275mm^2	16.9316mm^2	$9.7564 \pm 8.6717\text{mm}$	$3.2086 \pm 2.6467\text{mm}$

Table 3.2: *Reprojection error before and after bundle adjustment.*

3D scene reconstruction and rendering. In [DM97] a method is described to build a high dynamic range image from multiple exposures by recovering the characteristic curve of the imaging device (or film). This curve maps the optical density (absorbance of the film) to the logarithm of the product of the exposure time ΔT_e and the irradiance E. As with common devices and under normal circumstances this mapping is unique, the objective is to find a monotonic function g that maps the irradiance-exposure time product to the digital intensity value that is obtained from the sensor:

$$Z_{ij} = g\left(E_i \cdot (\Delta T_e)_j\right) \tag{3.18}$$

where Z_{ij} is the digital intensity value at a spatial pixel position (index i) measured with one of the exposure times $(\Delta T_e)_j$ with index j. With the true scene radiance, g and the exposure time known, the irradiance E_i can be determined. Unfortunately, it is difficult to measure the true radiance in a scene. However, in [DM97] g is approximated from a couple of images taken from a static scene with different exposure times and without moving the camera. After this procedure, g is known up to scale which is sufficient for the applications of TRIVIS.

Data sets obtained using the described mapping can be stored in arbitrary resolution (depending on the number of exposures) in the relative log luminance space ($log(E_i \cdot \Delta T_e)$ with $E_i \sim E$) and have the further advantage that there is less noise present due to averaging at a location i over a number of measurements j.

Figure 3.7 shows the relative log luminance to intensity value mapping approximated from 8 exposures (1, 2, 5, 10, 30, 80, 150, and 300ms) at 400 pixel locations distributed evenly over the images of the video cameras facing a scene with sufficiently large dark and bright areas. The mapping is determined for every color channel separately. The vertical shift is arbitrarily chosen and set equal for all cameras (intensity value 128 is assumed to have zero relative log luminance). For camera two it is observed that in dark areas the blue channel has a much stronger fall off than the green and red channel. Additionally, camera two seems to produce greater intensity values for dark areas than the other two cameras. White balancing is not performed as this would need a white (or at least known) color calibration pattern at a known position in the scene with well defined illumination.

3.3.2 Inter-camera color mapping

Determining the relative irradiance mapping of the cameras as described in the former section has two applications for TRIVIS. The first one is to be able to acquire high dynamic range image-based data sets. The second is the linearization of the response function of the

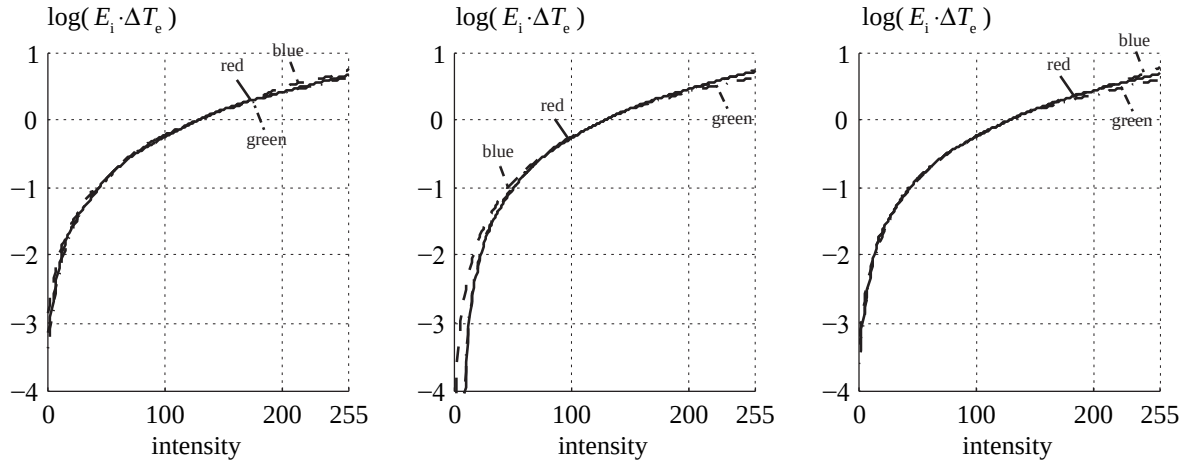

Figure 3.7: *The relative irradiance to intensity mapping of the cameras (cameras one to three from left to right).*

cameras. This linearization is particularly important to remove the inter-camera color mismatches. As described in, e.g., [JWV+05] a global error correction scheme can be applied to minimize the error between measurements obtained from different cameras imaging the same scene point. A color mapping between a reference camera and the other cameras is determined by estimating a 3×4 color conversion matrix L_{ij} which maps RGB values of camera i to RGB values in camera j where the source's RGB value vector is augmented with a fourth component equal to 1:

$$\begin{pmatrix} R_c \\ G_c \\ B_c \end{pmatrix} = L_{ij} \begin{pmatrix} R_o \\ G_o \\ B_o \\ 1 \end{pmatrix} \tag{3.19}$$

Here, subscripts "o" and "c" denote original values and corrected values, respectively. L_{ij} is fitted from correspondences on a colored calibration pattern (see Figure 3.9) in the cameras by solving the overdetermined linear system in Equation (3.19) in the least squares sense. The location and orientation of the color calibration pattern in the images is obtained manually. The pixels in the interior of the calibration pattern are transformed to a 100×100pixel area where at each pixel location the measurements of all three cameras can be used to estimate L_{ij}. For TRIVIS this mapping is determined in the log luminance space rather than in the intensity space. Figure 3.8 shows the original and the corrected inter-camera color mappings. The reference camera is chosen to be the second camera, and therefore the second camera's relative irradiance values are not altered. For the first and third camera, L_{12} and L_{32} are determined and used to manipulate the relative irradiance to resemble the relative irradiance to intensity mapping of the second camera. Figure 3.9 shows the result of the inter-camera calibration procedure in the RGB color space with an exposure of 45ms. This is done by inverting the function g to obtain intensity values from relative irradiance values using a fixed exposure time. In the left of Figure 3.9 the calibration pattern before color calibration is shown. Pixels from a 2×2 area are taken from different cameras to visualize the difference of the images of the same scene point in the three cameras. On the right side of the figure, the corrected calibration pattern is shown. The contrast of the images is enhanced for

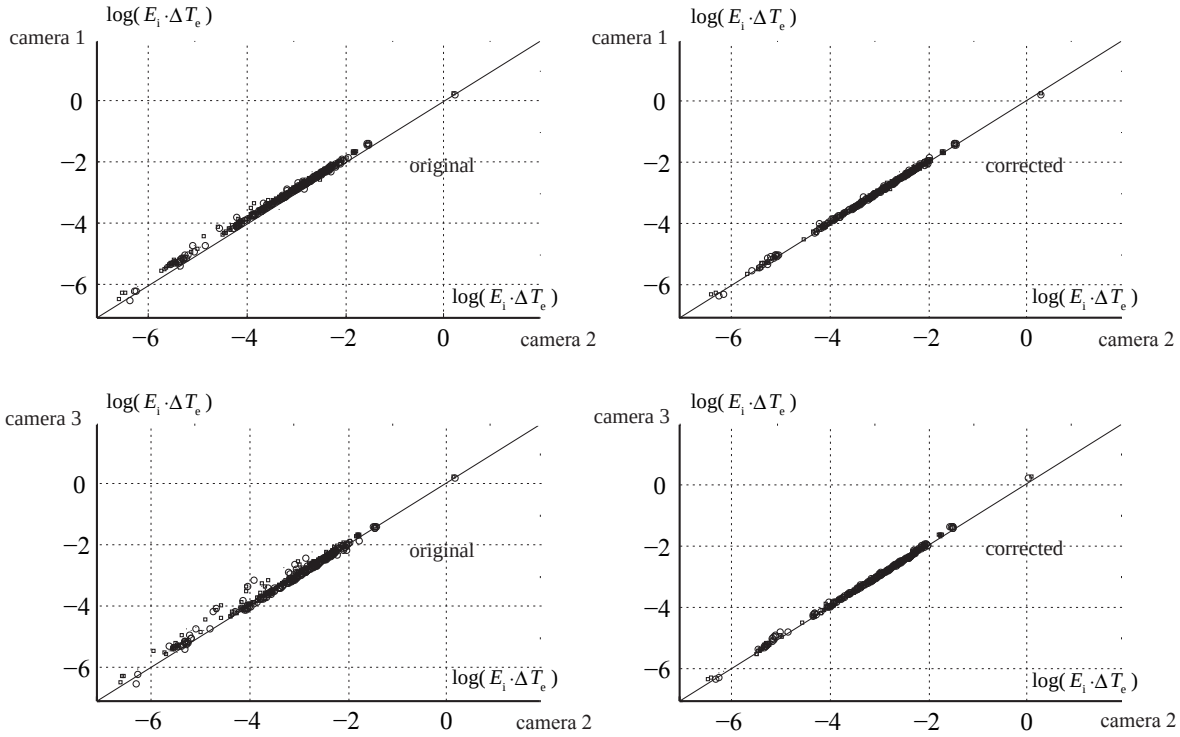

Figure 3.8: *The color mapping with respect to the second camera. Intensity values of the first camera (top) and the second camera (bottom) should match when looking at the same scene point (samples should lie on the bisector). Before calibration (left) and after calibration (right) in the log-luminance space.*

clearer presentation. Table 3.3 finally gives numerical results for the calibration procedure. The root mean squared error (RMS) in the intensity value domain (0-255) of the first and third camera with respect to the reference camera before and after calibration are shown for each color channel separately. The RMS drops by a factor of four. The maximum error also decreases significantly. The remaining errors are due to sensor noise and non linear effects that are not captured by the mappings. Comparable results (for a much larger camera array) have been reported in [JWV+05].

	RMS			max. error		
	R	G	B	R	G	B
gain set equal for all cameras	9.9	11.2	14.3	34.5	28.6	38.6
after color calibration	2.7	2.2	2.5	11.6	10.7	12.8

Table 3.3: *Matching error in the interior of the calibration pattern (Figure 3.9) in the red, green and blue components before and after color calibration with respect to the second camera.*

Due to the very limited field of view of the cameras, a vignetting effect has not been taken into account during color calibration. With the described procedure, high dynamic range imaging can be performed, but for single exposure acquisition, inter-camera color calibra-

tion can be obtained via the relative irradiance space and the L_{ij} matrices and finally transforming back to RGB intensity space with the inverse of the monotonic function g. Results for rendering with high dynamic range using TRIVIS are given in Figure 3.24 on page 72.

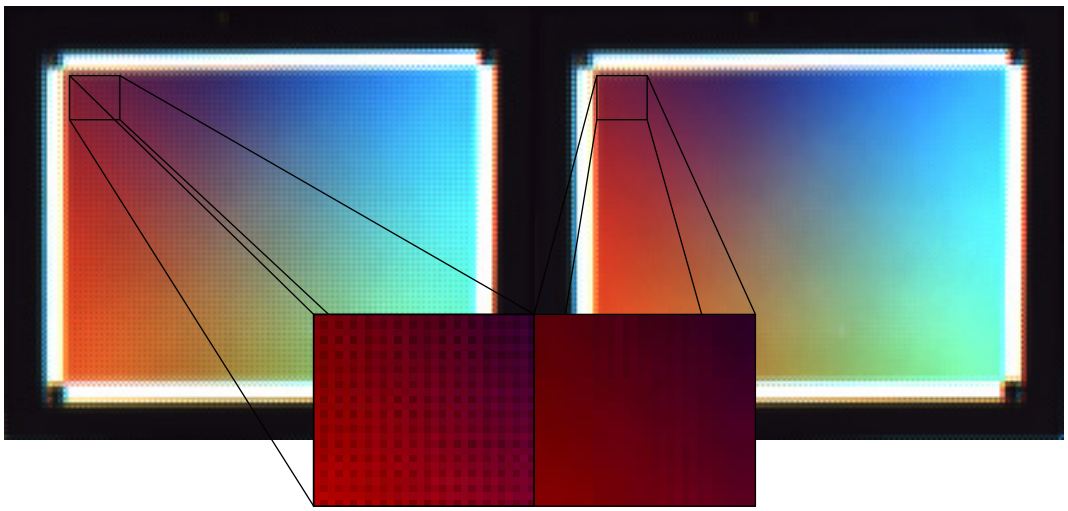

Figure 3.9: *Pixels in a 2×2 image region are taken from different cameras. The upper left pixel is taken from camera 1. The upper right and lower left pixels from cameras 2, and the lower right pixel is taken from camera 3. (left) Before color calibration and (right) after color calibration. For the magnified areas the contrast has been enhanced for clearer presentation.*

3.4 Pose estimation using TRIVIS

Scene acquisition using TRIVIS is performed by moving the acquisition device in front of the scene of interest. This is done manually and for the resulting series of laser scans and image triplets (hereafter called "shots"), the problem of estimating the pose of the device for every captured shot with respect to a world frame remains. When the pose of the camera for every input image is known, an unstructured image-based scene representation as discussed in Section 2.3.2 on page 14 is obtained.

3.4.1 Overview

Since the intrinsic parameters and device extrinsics of TRIVIS are known (see Section 3.2 on page 35), any two corresponding points in the images of any two of the three rigidly mounted cameras can be used to reconstruct a 3D point with respect to device coordinates by triangulation. This property of the fully calibrated acquisition system is exploited for pose estimation by matching subsequent shots via their partial and sparse 3D scene reconstruction. I.e., the relative rotation and translation of TRIVIS between any two exposures is obtained by first

- extracting a number of 3D points for each of the shots via two-point correspondences within the images of one shot, then

- for every 3D point in one shot a corresponding 3D point in the other shot is found by establishing two-point correspondences in images between both shots, and then,
- the relative rotation and translation of the two device coordinate frames is estimated by minimizing a distance measure between the two corresponding 3D point clouds.

Basically, for two sets of corresponding 3D points, the transform that relates them is to be found. Such a problem is commonly referred to as direct pose estimation (or absolute orientation, hand-eye transform etc.). In [Hor87] a closed-form solution in the least squares sense is presented that can be used with at least three 3D point correspondences.

With such an estimate of the relative pose of two subsequently taken image triplets, the whole sequence of acquired images can be registered by defining, e.g., the device coordinate frame of the first shot to be the world frame. But, as pose estimation errors accumulate along the motion trajectory of the acquisition device, it is important to track features over more than two subsequent shots. With point correspondences in multiple images a more accurate overall solution of the pose estimation problem can be found. For the acquisition procedure considered in this chapter this strategy is also pursued, but the advantages of having a precalibrated system are exploited to increase stability and accuracy.

3.4.2 Intra and inter shot feature matching and sparse reconstruction

The very first step for relating images is to extract suitable features in the images. For the TRIVIS system, point features are chosen that can be extracted as described in Appendix A.1 on page 159. The search for corresponding features among the images of one shot can be simplified as the relative positions of the cameras are known. Then, due to the epipolar geometry [LH81], the search for the correspondence of a feature detected in one image is constrained to a line in each of the other two images. Further, the corresponding 3D points are also constrained to be in a certain distance range with respect to the corresponding cameras. These constraints result in a limited, rectangular search area which is determined by three parameters through the device calibration and two view geometry (see [HZ04]):

- $\Delta d_{m,max}$ is the maximum allowed distance (in image coordinates) between the projection of a possible corresponding feature from the epipolar line (determined by the camera calibration and feature position for which a correspondence is to be found),
- $Z_{m,min}$ is the minimum allowed distance of a 3D feature from the cameras, and
- $Z_{m,max}$ is the maximum allowed distance of a 3D feature from the cameras.

Intra shot feature matching is performed on all acquired shots of a sequence separately using the method described in Appendix A.1 on page 159 using a window size of B_m. The obtained correspondences are triangulated (the non-linear method for triangulation from multi-view correspondences from [HZ04] is used) to obtain estimated 3D points in the respective device coordinates and the observation of their projections. This "guided matching" approach not only speeds up the sparse 3D reconstruction by constraining the search area, but also makes the resulting correspondences more reliable than in the case for uncalibrated single camera systems.

To relate two shots to each other, point correspondences between the images of different shots have to be established. As no a priori information about the motion of the device is available, related image regions have to be detected using a full search. However, images

captured by the same camera in subsequent shots are likely to have largely overlapping areas. Therefore, for two subsequent shots, only three feature matching passes are performed, one for each camera. The obtained correspondences are used to link the 3D points from the intra shot feature matching step between the shots (see Figure 3.10).

3.4.3 Direct pose estimation

In contrast to common structure from motion and self-calibration techniques, direct pose estimation requires a precalibration of the device to obtain 3D points and their correspondence information as described in the former sections. In the noise free case and without mismatches, the problem of estimating the relative position and orientation between two shots can be formulated as:

$$\min_{\boldsymbol{\Delta R},\boldsymbol{\Delta T}} \sum_{k=1}^{|\boldsymbol{P}|} \|\boldsymbol{p}_k - (\boldsymbol{\Delta R} \cdot \boldsymbol{q}_k + \boldsymbol{\Delta T})\|^2 \tag{3.20}$$

where $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ are 3-vectors representing corresponding points in device coordinates out of the point sets $\boldsymbol{P}$ and $\boldsymbol{Q}$ of 3D points in the first and second shot, respectively. The 3×3 rotation matrix $\boldsymbol{\Delta R}$ and the 3-vector $\boldsymbol{\Delta T}$ define the rotation and translation between the point clouds. In [Hor87], a scale is determined which is ignored here due to the fact that only rotation and translation can occur during motion of the acquisition device. While $\boldsymbol{\Delta T}$ is easily determined by the difference of the centroids of the two point sets, the rotation needs the correspondence information for closed-form formulation (see [Hor87] for details on the implementation). Figure 3.10 illustrates the direct pose estimation approach for the three camera system. Image point correspondences in any two camera images of each shot are triangulated and corresponding 3D points are obtained. Image point correspondences between images across the shots link these 3D points. The pose of the right shot is then determined with respect to the arbitrarily chosen world frame that coincides with the device coordinate frame of the first (left) shot.

In practice, outliers from the intra and inter shot feature matching steps are present in the input to the direct pose estimation procedure. To handle these outliers, a robust algorithm is used. The RANSAC [FB81] algorithm is fed with subsets of the linked 3D points of the two shots that are to be registered. The solution to (3.20) that includes the most point correspondences of the whole set of 3D points that satisfy:

$$\Delta Z_{rm,max} > \|\frac{\boldsymbol{p}_k - (\boldsymbol{\Delta R} \cdot \boldsymbol{q}_k + \boldsymbol{\Delta T})}{p_k^3}\| \tag{3.21}$$

is taken. Here, the parameter $\Delta Z_{rm,max}$ is the maximum allowed relative deviation with respect to the coordinate frame of the previous shot and is set to 0.05 (50mm) in the experiments. The denominator p_k^3 is the Z-coordinate of the 3D point $\boldsymbol{p}_k$ in the first shot's coordinate system and normalizes the distance of the 3D points to accommodate to the fact that triangulation in the presence of noise is more inaccurate for 3D points that are farther away from the device coordinate centers. $\boldsymbol{\Delta R}$ and $\boldsymbol{\Delta T}$ are subsequently refined using (3.20) with all of the points for which (3.21) holds (i.e., the inliers). The obtained solution minimizes the mean squared geometrical distance between corresponding points in the triangulated point clouds.

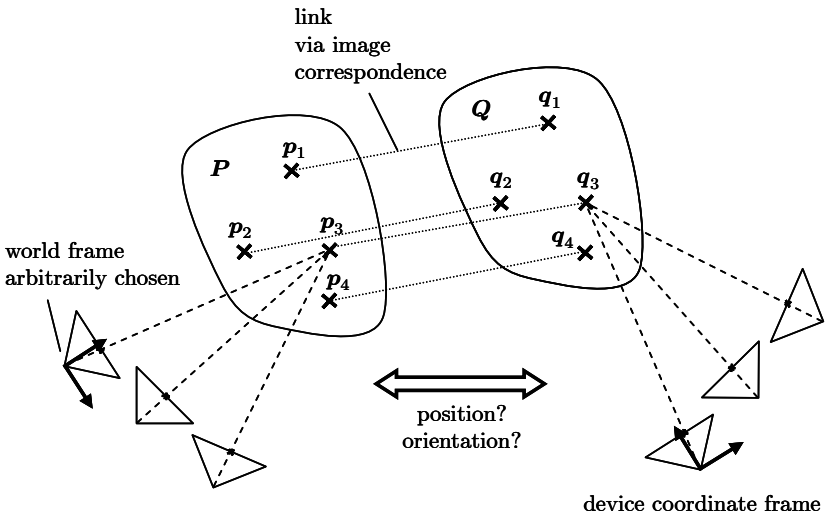

Figure 3.10: *Direct pose estimation simplified in a 2D illustration. 3D scene points reconstructed from the three images in every shot are linked by image correspondences between two shots. The point clouds are matched as described in the text.*

Altogether, the procedure for approximating the pose of subsequent shots consists of three guided feature matching passes for any two images of each shot and three unconstrained feature matching passes to link the 3D points between the shots. With these six feature point matching passes, scene points may already have linked point correspondences in up to six images. Due to the closed-form solution to recover relative translation and rotation between subsequent shots, pose estimation is quite efficient.

3.4.4 Feature tracking and merging for long sequences

The relative pose estimation procedure discussed in the former section can be applied and accumulated for all subsequent shots that have been triggered during acquisition of a scene to obtain an approximation of their absolute pose. However, to capture a dense and approximately critically sampled scene representation it is required to move TRIVIS in such a way that the motion trajectory eventually crosses itself, or at least shots far apart in time are likely to be spatially near. In this case, pose estimation on subsequent shots introduces accumulated errors that increase dramatically as the acquired sequences become long.

To overcome this problem, shots are inserted into the current state of the scene representation one after the other and are registered to a couple of shots near the current one instead of only considering a single previously taken shot. This strategy was introduced in [HPDvG99] for a single (uncalibrated) camera system. By finding a subset of the already registered images that are most likely to contain scene content which is also visible in the current image, error propagation is handled properly, even for large sequences.

For TRIVIS the first step is to choose one shot whose device coordinate frame is defined to be the world frame. Here, the first shot of the sequence is chosen and the next captured shot is registered using the direct pose estimation procedure. The rotation $\boldsymbol{R}_j$ and translation $\boldsymbol{T}_j$

of the device in shot j with respect to the world frame is determined by

$$\boldsymbol{R}_j = \Delta\boldsymbol{R}\boldsymbol{R}_{j-1} \quad \text{and} \quad \boldsymbol{T}_j = \Delta\boldsymbol{T} + \boldsymbol{T}_{j-1} \tag{3.22}$$

while the rotation $\boldsymbol{R}_{ij}$ and translation $\boldsymbol{T}_{ij}$ of camera i in shot j with respect to the world frame is determined by

$$\boldsymbol{R}_{ij} = \boldsymbol{R}_i\boldsymbol{R}_j \quad \text{and} \quad \boldsymbol{T}_{ij} = \boldsymbol{T}_i + \boldsymbol{T}_j. \tag{3.23}$$

Here, $\boldsymbol{R}_{j-1}$ and $\boldsymbol{T}_{j-1}$ are the absolute translation and rotation of the previous shot. $\Delta\boldsymbol{R}$ and $\Delta\boldsymbol{T}$ are the relative rotation and translation determined by direct pose estimation and $\boldsymbol{R}_i$ and $\boldsymbol{T}_i$ denote the relative rotation and translation of the cameras with respect to the device coordinate system (see Section 3.2.2 on page 36).

The image features that survived the RANSAC selection are merged via their intra and inter shot links by associating them with the 3D point $\boldsymbol{m}_k = (\boldsymbol{p}_k + (\boldsymbol{\Delta R} \cdot \boldsymbol{q}_k + \boldsymbol{\Delta T}))/2$ which is the average of the corresponding 3D points in both shots after registration. At this stage, for all cameras i in all shots j that are registered so far, projection matrices $\boldsymbol{P}_{ij}$ in the form of (3.2) can be determined. With the 3D features, a sparse approximation of the scene structure is also available. Using this information, additional matches in cameras near a camera of an inserted shot are found to increase the number of multi-view features. Such cameras are also selected according to their spatial distribution, i.e., a fixed number of cameras (N_m=10) are chosen that are within a distance ΔT_{max} and ideally equally distributed around the current camera. Figure 3.11 (left) shows an exemplary situation where for shot j features found in shot $j-3$ and $j-4$ that got out of sight during motion of the acquisition device along the motion trajectory, can be rematched.

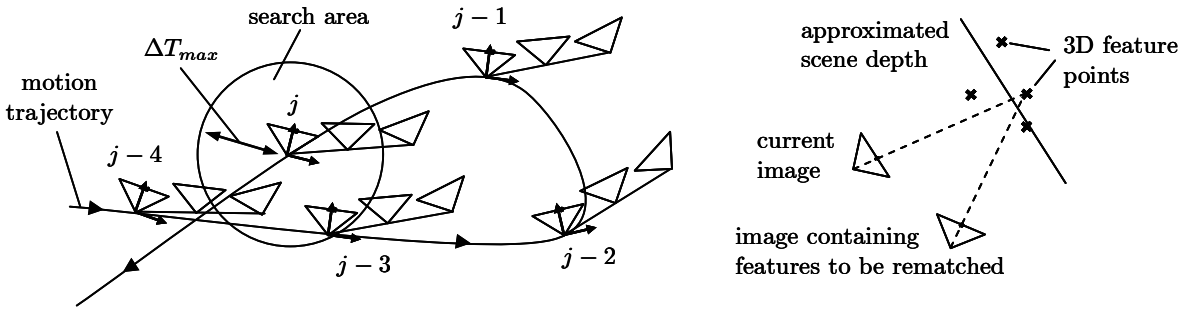

Figure 3.11: *Rematching to prevent error propagation during sequential matching. (left) Image triplets are captured along a motion trajectory that crosses itself. Features that got out of sight but are seen in one camera of shot j are rematched and merged with features seen in cameras of shot $j-3$ and $j-4$ within the distance ΔT_{max}. (right) Approximation of the scene structure for image warping during rematching.*

To handle heavy translational motion and rotation, the current 3D structure of the model is used to warp the image texture and feature coordinates of the camera image that is to be rematched into the current camera. The scene structure is approximated by fitting a plane to the 3D scene points as shown in Figure 3.11 (right). The guided matching approach from Section 3.4.2 is used with $\Delta d_{m2,max}$ which replaces $\Delta d_{m,max}$ and is set to 20 pixels to constrain the search area. 20 pixels seems to be large, but there might already be a significant

error due to sequential matching. The rematched features are then merged with the current list of features and corresponding 3D point estimates.

Up to now, only minimization of the geometrical error rather than the reprojection error of the merged 3D points in the already registered images has been considered. A sparse bundle adjustment step is now performed (see Section A.3 on page 160) for every new inserted image in order to minimize

$$\sum_{i=1}^{|\boldsymbol{W}|}\sum_{j=1}^{3}\sum_{k=1}^{|\boldsymbol{M}|}\left\|\boldsymbol{x}_{ijk}-\boldsymbol{K}_i\left(\left(\boldsymbol{R}_i\boldsymbol{R}_j\right)^T\Big|-\left(\boldsymbol{R}_i\boldsymbol{R}_j\right)^T\left(\boldsymbol{T}_i+\boldsymbol{T}_j\right)\right)\cdot\boldsymbol{m}_k\right\|^2 \tag{3.24}$$

and therefore to minimize the distance between the reprojections of the 3D features and their measurements in the images. Here, $\boldsymbol{W}$ is the set of shots under consideration (already registered ones plus the current). $\boldsymbol{M}$ is the set of 3D features $\boldsymbol{m}_k$, $\boldsymbol{x}_{ijk}$ is the measured projection in image i of shot j for point $\boldsymbol{m}_k$, and $\boldsymbol{K}_i$ is the intrinsic calibration matrix for camera i obtained during the precalibration procedure. The variables that are altered by the sparse bundle adjustment algorithm are the scene points $\boldsymbol{m}_k$, the absolute device pose $\boldsymbol{T}_j$ and orientation $\boldsymbol{R}_j$ with respect to the world frame. For 3D points $\boldsymbol{m}_k$ that do not have a measurement $\boldsymbol{x}_{ijk}$, the reprojection error is set to zero. The device intrinsics and extrinsics remain untouched during this optimization step. Only the device pose for every shot is subject to changes. This optimization step usually converges after as few as five iterations of the bundle adjustment procedure.

3.4.5 Global refinement

Once all shots are incorporated into the scene representation, an overall refinement pass is performed to optimize the device calibration, pose estimation, and sparse scene structure. The expression that is minimized is equivalent to (3.24) whereas for the global refinement also the device extrinsics are subject to modifications by the bundle adjustment algorithm. The intrinsic parameter matrices $\boldsymbol{K}_i$ remain untouched during optimization. Multiple passes of this refinement step are performed. After each pass the fraction Δr_m of the measurements showing the largest reprojection error are removed and the minimization is reinitialized. The optimization terminates when either the total mean reprojection error falls below a threshold r_m or the change in the parameters is too small. During the optimization a minimum number k_m of features per shot is kept. The global optimization step usually terminates after as much as 200 iterations of the bundle adjustment procedure and in altogether usually 5 passes with outlier removal.

This last optimization step alters the device extrinsics, and therefore, also alters the relative position of the laser scanner with respect to the cameras. To overcome this drawback, (3.17) is minimized again, this time using the intra shot feature points from the acquired sequence for the cameras and keeping the laser-camera correspondences from the calibration procedure to find the relative position of the laser scanner by keeping the cameras' intrinsic parameters and the device extrinsics untouched. Early experiments showed that due to the scanners inaccuracies, the overall change of the scanner position and orientation is minimal and thus considered negligible.

Table 3.4 summarizes the pose estimation algorithm while Table 3.5 shows the parameters that can be adjusted as well as their values used in the subsequent experiments.

- Extract point features from all images in the input image set (see Appendix A.1 on page 159).
- Match features across the three images of each shot and triangulate them to obtain a set of 3D features and their corresponding measured projections using guided matching.
- Establish feature correspondences between images of the same camera across all subsequently taken shots.
- Starting with the first shot, successively add the next taken shot and estimate its relative pose using direct pose estimation. From the current structure and motion estimate, determine a number of cameras that are used for rematching in the spatial neighborhood of the cameras of the current shot. Warp the rematching images into the current camera using a surface approximation and perform guided rematching. Optionally, refine the current structure and motion estimate by a bundle adjustment.
- Globally refine the structure and motion parameters as well as the device extrinsics.
- Recompute the device extrinsics with fixed camera intrinsics and extrinsics to obtain the scanner calibration.

Table 3.4: *Summary of the pose estimation algorithm.*

3.4.6 Results

The pose estimation algorithm described in the former section is tested on two real world sequences. The first sequence consists of 270 images in 90 shots with forward motion, sidesteps, and rotation. The second sequence consists of 900 images from 300 shots with free translation and rotation as it would be performed for the acquisition of an image-based scene representation that approximates a 2D light field captured on a part of a hemisphere.

Figure 3.12 (top) shows three images (camera 1) of the 270 image sequence of a hybrid outdoor and indoor scene. The images are taken from shots 1, 25, and 80, respectively. In the figure, some of the joint features of the three shots are marked. Note the scaling through the images due to forward motion of the device.

Figure 3.12 (bottom) shows three images (camera 3) of the 900 image sequence (300 shots) of an indoor scene (though naturally illuminated through the window). The images are taken from shots 10, 180, and 300, respectively. In the figure, some joint features of the three images are marked. Note the heavy rotation of the camera. Though the considerable amount of scaling and rotation of the device during acquisition, the image warping procedure described in

B_m	window size for matching features among the camera images	5 px
$\Delta d_{m,max}$	allowed maximum distance of an image feature position from the epipolar line (intra shot matching)	3 px
$\Delta d_{m2,max}$	allowed maximum distance of an image feature position from the epipolar line (inter shot matching)	20 px
$Z_{m,min}$	minimum depth of 3D features w.r.t. the device frame	0.5m
$Z_{m,max}$	maximum depth of 3D features w.r.t. the device frame	70m
$\Delta Z_{rm,max}$	relative deviation allowed for inlier detection during direct pose estimation (w.r.t. Z=-1m)	0.05
Δr_m	maximum fraction of image measurements that are removed after each iteration of the global optimization pass	0.02
ΔT_{max}	radius within camera poses are searched for rematching	150mm
N_m	maximum number of neighboring images that are used for rematching	10
r_m	the global optimization procedure stops when the mean reprojection error falls below this value	1 px
k_m	minimum number of matches that are kept for every image	20

Table 3.5: *Summary of the parameters and settings for the pose estimation algorithm*

Section 3.4.4 makes it still possible to keep feature tracking reliable.

Figure 3.13 shows the motion trajectories and the obtained point clouds after the registration of the sequences. The monitor and keyboard are sketched for clearer presentation. Figure 3.14 shows the track matrix of the two sequences which denotes the features that are visible in the corresponding shot. For the 90 shot sequence on the left it is observed that some shots share a large number of features while there is a clear break between some groups of shots. This is because the maximum distance between cameras to be candidates for rematching is too small to capture more images and therefore feature rematching stops. This could be solved by increasing ΔT_{max}. However, the number of feature matching passes would increase significantly. For the 300 shot sequence it is observed that for almost all shots, the rematching procedure finds images that have been captured at significantly different time instances. This is because the device motion was rather random during acquisition resulting in many crossings and shots that come close to each other. Note that most of the features that are visible over the whole sequence are initialized in the first few frames.

Figure 3.15 shows the "track length probability", which is the actual fraction of features visible in a certain number of images. On the left, this probability is shown for the 90 shot sequence. Some features are tracked in over 118 images. The diagram shows only the feature track length of more than 8 successfully tracked image correspondences for clearer presentation. The same applies to the track length probability for the 300 shot sequence. There, due to the object centered way of recoding shots, the most popular feature is visible in 580 (of 900) images.

Figure 3.16 shows the number of features per view for the two test sequences. For the 90 shot sequence the number of features per image is balanced between 60 and 80. For the 300 shot sequence it happens that the number of tracked features in the images dramatically drops. This is mainly due to the partially textureless scene surfaces (compare to the middle

image in the bottom of Figure 3.12). In such regions, image one or two of the three images have to "support" the third image to be correctly registered. Conventional feature tracking methods using only one camera would loose track in these areas.

Figure 3.17 (left) shows the reprojection error after pose estimation. For both sequences the reprojection error is well balanced between the cameras (not shown in the figure, but it can be deduced from the homogeneous distribution). Table 3.6 gives numerical results for the 300 shot sequence (mean reprojection error and the standard deviation in pixels) before and after global optimization. The significant improvement is mainly due to removed outliers.

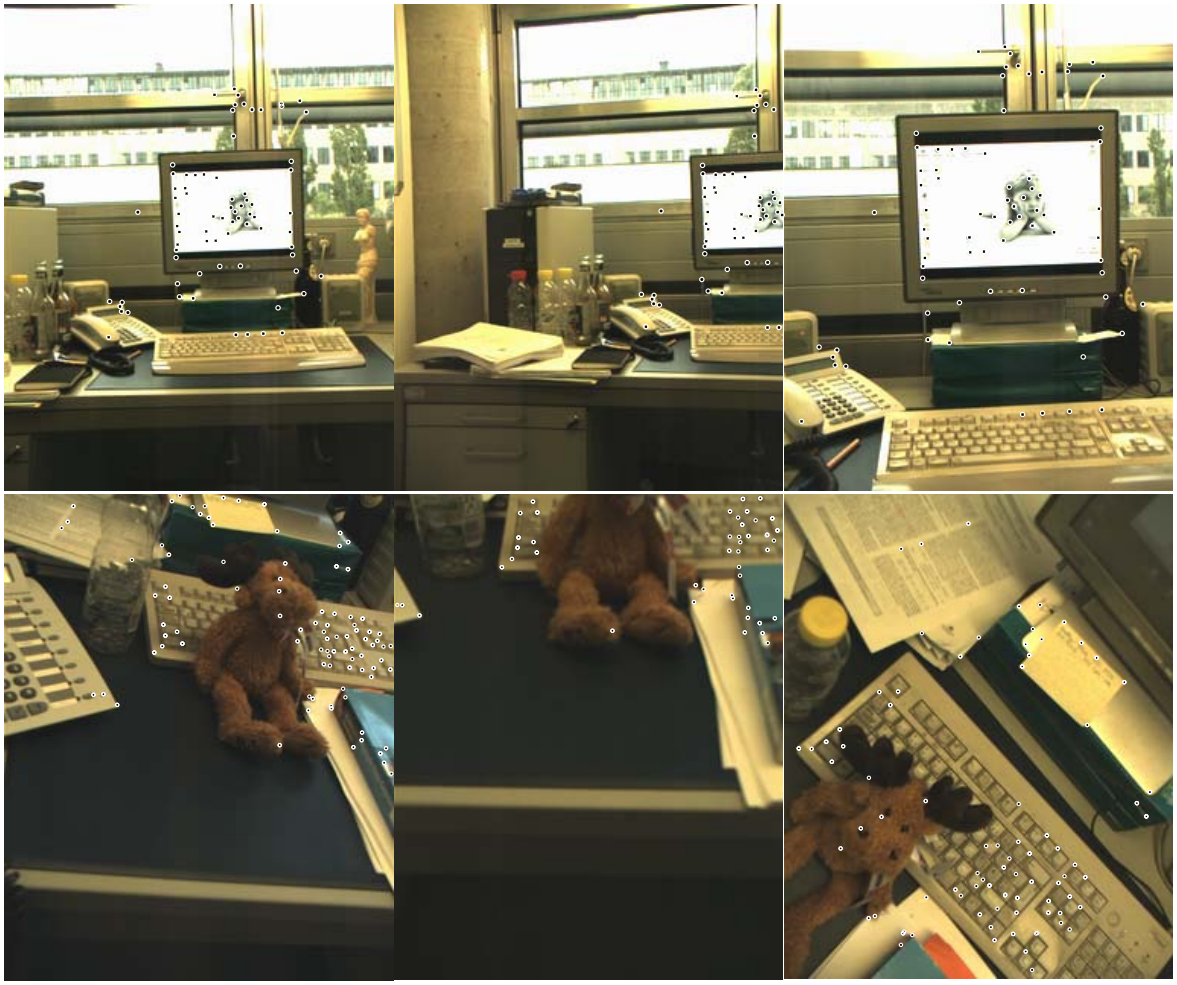

Figure 3.12: *(top) Images of camera 1 in the shots 1, 25, and 80 of the 90 shot test sequence consisting of forward and backward motion as well as sidesteps. (bottom) Images of camera 3 in the shots 10, 180, and 300 of the 300 shot test sequence consisting of all possible motion. Some features that have been tracked over these shots are shown.*

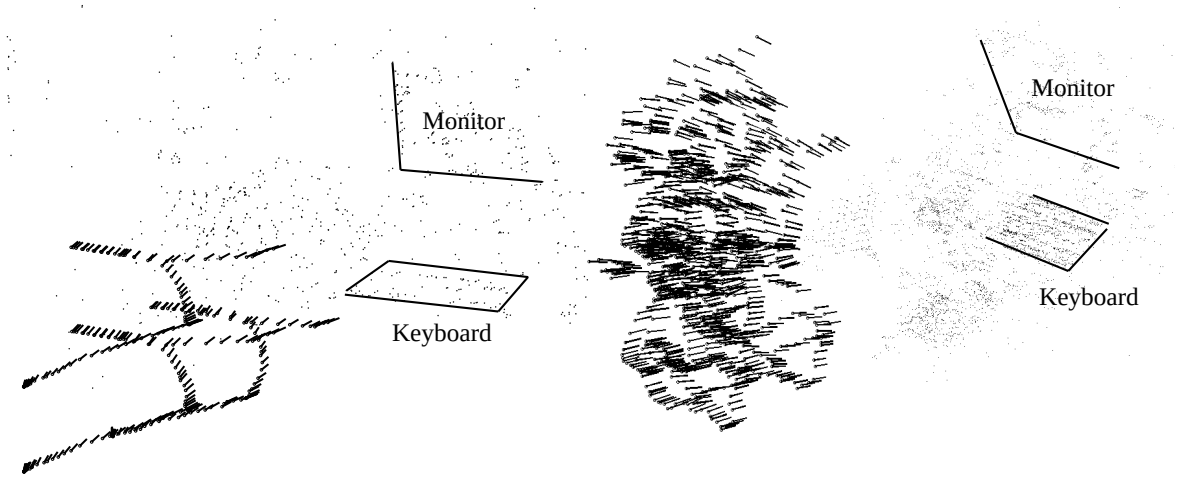

Figure 3.13: *(left) Camera centers and viewing direction together with the obtained point cloud for the 90 shot sequence. (right) Camera centers and viewing direction together with the obtained point cloud for the 300 shots sequence. The monitor and keyboard are sketched for clearer presentation.*

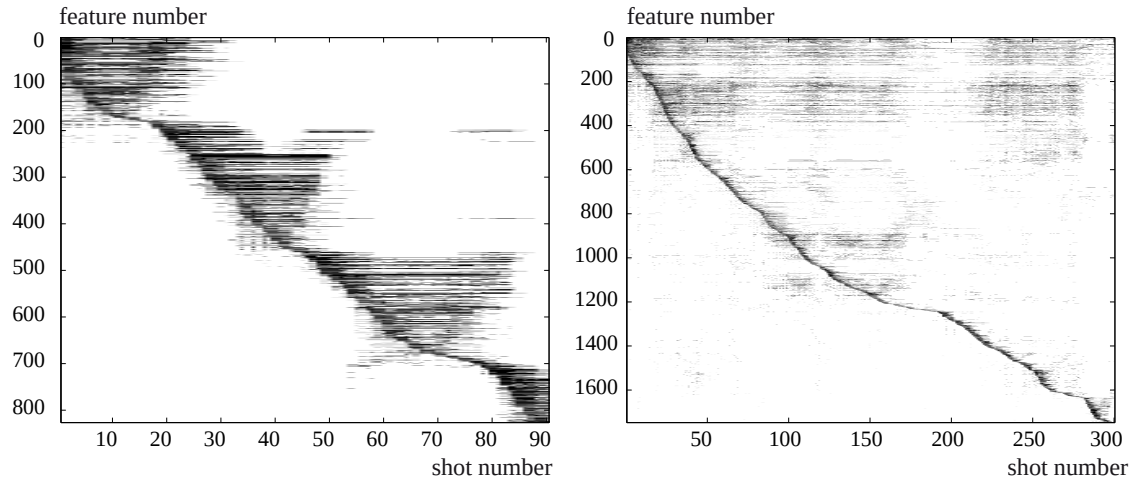

Figure 3.14: *Feature-shot trackmatrix for the (left) 90 shot sequence and the (right) 300 shot sequence. Dark dots denote that the corresponding feature is visible in at least one of the images of the shot.*

3.5 Local scene geometry reconstruction

As discussed in Sections 2.3.2 and 2.4.2 on pages 14 and 18, respectively, even with undersampled scene representations photorealistic rendering can be achieved when a geometric model is available. This is true under the assumption that the scene is Lambertian and the geometric model is sufficiently accurate. Rather than extracting a single common scene geometry model, in this section the reconstruction of multiple local models, each valid when used for rendering from within a small region of the viewing space spanned by the input camera locations, is discussed. These models are intended to also model non-Lambertian

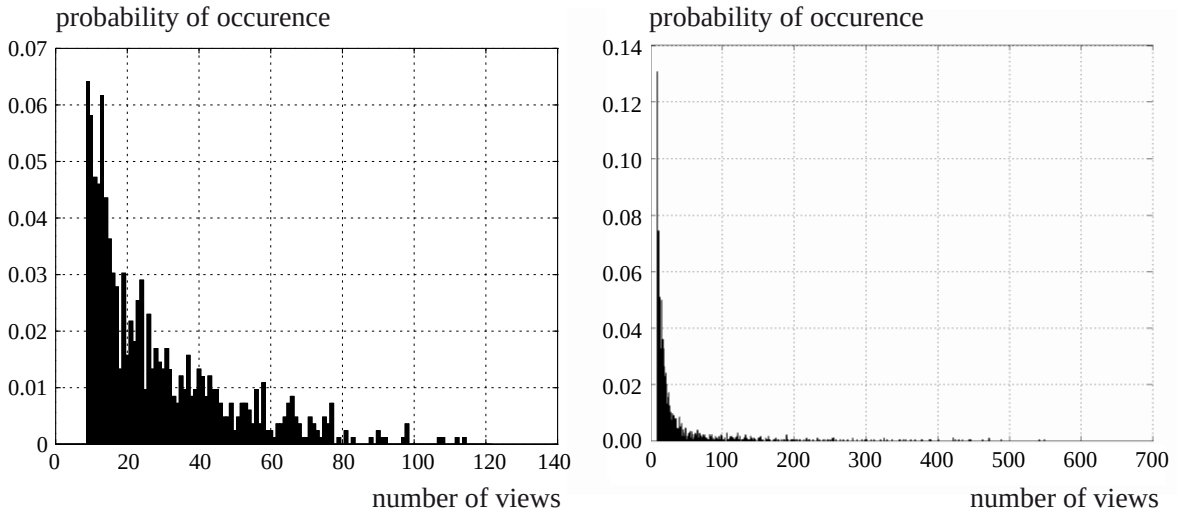

Figure 3.15: *(left) The probability for a feature to be visible in a specific number of image for the 90 shot sequence. (right) The probability for a feature to be visible in a specific number of image for the 300 shot sequence.*

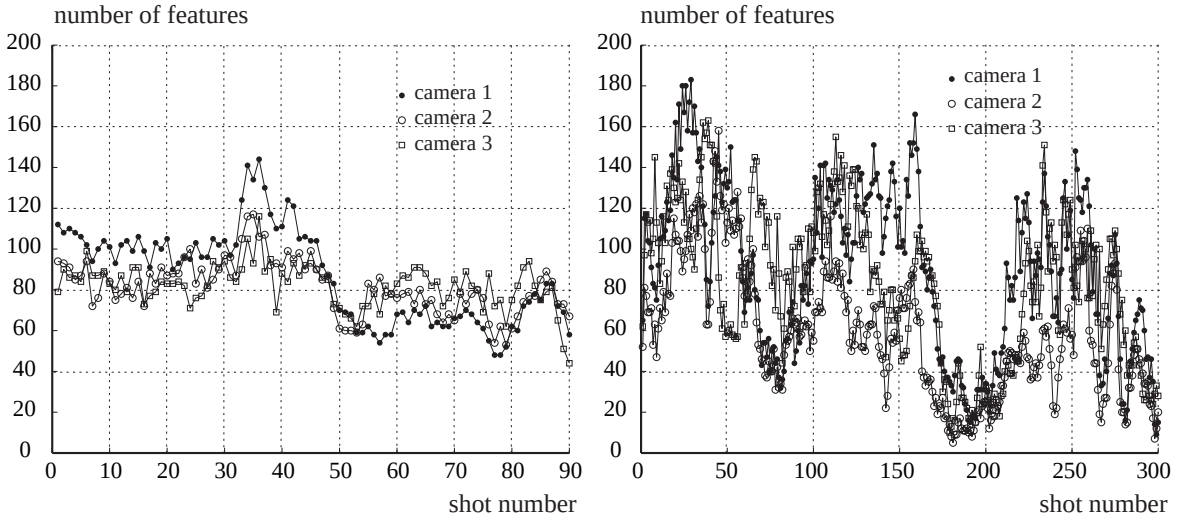

Figure 3.16: *(left) The number of features found and tracked in the cameras during acquisition of the 90 shot sequence. (right) The number of features found and tracked in the cameras during acquisition of the 300 shot sequence.*

effects like specularities as these effects can be regarded as geometric entities with a virtual depth [TSK+01, KS04]. The local models are represented by dense per pixel depth maps for each of the input images. In the case of rendering with TRIVIS, the local geometric models have to be extracted from the input data itself. An algorithm that fuses reconstructed depth information from classical multi-view stereo and active measurements from the laser scanner is discussed in this section. The main issues summarize as follows:

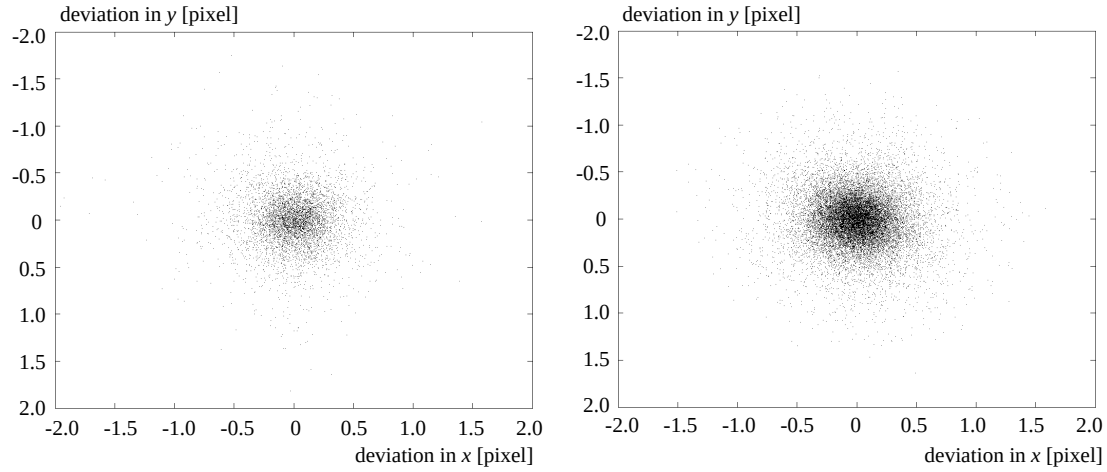

Figure 3.17: *(left) The reprojection error after global optimization for 25% of the point measurements for the 90 shot sequence. (right) The reprojection error after global optimiztion for 25% of the point measurements for the 300 shot sequence.*

	sequential	global
camera 1	$0.960 \pm 0.677\,\text{px}$	$0.304 \pm 0.227\,\text{px}$
camera 2	$1.194 \pm 0.773\,\text{px}$	$0.325 \pm 0.236\,\text{px}$
camera 3	$1.097 \pm 0.714\,\text{px}$	$0.329 \pm 0.232\,\text{px}$
all	$1.067 \pm 0.721\,\text{px}$	$0.318 \pm 0.231\,\text{px}$

Table 3.6: *Reprojection error before and after global optimization (mean and standard deviation) for the 300 shot sequence.*

- **Noise.** Sensor noise, blurring, varying lighting conditions, and inaccurate calibration have to be taken into account.
- **Untextured regions.** While it is not possible to establish correspondences in untextured regions using visual sensors, active measurement systems using, e.g., a laser beam can still provide reliable estimates.
- **Depth discontinuities and occlusions.** Flat areas and depth discontinuities should be preserved.
- **Physical limitations.** While the laser scanner provides a low spatial resolution and samples often are subject to systematic errors, imaging systems allow for a dense sampling of intensity values.

The main algorithm is based on global energy minimization. A cost function is used that consists of a local data term that describes the matching cost with respect to, e.g., color consistency and measurements from the laser scanner. Another term of the cost function is intended to preserve smoothness within the depth images by taking the possible depth values of neighboring pixels into account. The cost function is then minimized by an approximation algorithm based on belief propagation adapted from common stereo reconstruction research [SZS03, FH04].

3.5.1 Sensor data fusion

Three depth cues are used for local scene geometry reconstruction:

- The depth information provided by the 3D features that survived the pose estimation step.
- The depth information provided by the laser scanner.
- The depth reconstructed from common multi-view stereo.

These cues differ in the reliability of their measurements and sample density. While the 3D features are reliable as they undergo a robust selection and often are generated from correspondences in a large number of images, they only provide a very sparse reconstruction. The depth measurements from the laser scanner provide a relatively high resolution in depth, but are also sparsely sampled in space when compared to the sample density of the intensity values from the cameras. Further, laser scanner data is sampled with a much higher temporal sampling frequency than the camera images which actually provide the pose information. This makes measurements accurate only at sampling times that are close to the sampling times of the images. The pose of the other laser scans has to be interpolated between every two shots taken from the cameras. Multi-view reconstruction provides the sample density that is required for reconstruction, but often produces outliers because of non-Lambertian effects like specularities and occlusions/disocclusions, inaccurate inter-camera color calibration, or within untextured areas. Pose estimation inaccuracies have an impact on all depth cues.

To incorporate all the information that is available from the sensors, even if it is sparsely sampled, the general framework for dense depth estimation is defined as follows. For a set $\boldsymbol{P}$ of pixels and a set $\boldsymbol{D}$ of depth labels, find the labeling $\boldsymbol{F}$ that assigns a label $d_p \in \boldsymbol{D}$ to each pixel $\boldsymbol{p} \in \boldsymbol{P}$ so that the global energy

$$E(\boldsymbol{F}) = \sum_{\boldsymbol{p} \in \boldsymbol{P}} C_d(\boldsymbol{p}, d_p) + \sum_{(\boldsymbol{q}, d_q) \in \boldsymbol{N}} V(\boldsymbol{p}, \boldsymbol{q}, d_p, d_q) \tag{3.25}$$

is minimized. Here, $\boldsymbol{q}$ is a pixel in a local neighborhood $\boldsymbol{N}$ of $\boldsymbol{p}$. $C_d(\boldsymbol{p}, d_p)$ is the data cost for assigning a depth corresponding to depth label d_p to pixel $\boldsymbol{p}$. V describes the disparity cost of two neighboring pixels having different depths.

The data cost function

$C_d(\boldsymbol{p}, d_p)$ is split into three terms that represent the intensity matching cost C_i, the scanner matching cost C_s, and the 3D feature matching cost C_f:

$$C_d(\boldsymbol{p}, d_p) = C_i(\boldsymbol{p}, d_p) + \lambda_s \cdot C_s(\boldsymbol{p}, d_p) + \lambda_f \cdot C_f(\boldsymbol{p}, d_p) \tag{3.26}$$

Here, λ_s is a regularization parameter to weight passive and active geometry retrieval while λ_f weights the data cost term for the 3D information obtained during pose estimation. λ_s and λ_f are set to zero whenever no depth estimate is available, otherwise it is set to a value that corresponds to the reliability of the corresponding estimate as described later.

The intensity matching cost function uses a set $\boldsymbol{S}_I$ of images in which a pixel $\boldsymbol{p}$ in image I_{ref} is most likely to be visible (Section 3.5.2 on page 64 describes the selection of images in

$\boldsymbol{S}_I$ for rendering with TRIVIS):

$$C_i\left(\boldsymbol{p}, d_p\right) = \frac{1}{|\boldsymbol{S}_v|} \sum_{r \in \boldsymbol{S}_v \subseteq \boldsymbol{S}_I} \text{minimum}\left(\left(I_{ref}\left(\boldsymbol{p}\right) - I_r\left(\boldsymbol{p}, d_p\right)\right)^2, c_{max}\right). \tag{3.27}$$

Here, the function $\text{minimum}(a, b)$ takes on the value of the smaller value of a and b. The subset $\boldsymbol{S}_v$ among the used support images $\boldsymbol{S}_I$ is chosen such that the absolute difference in intensity at the distance determined by the depth label d_p between a pixel $\boldsymbol{p}$ in the reference image and the corresponding pixel in the support image is below a threshold c_{max}. $\boldsymbol{S}_v$ is forced to contain at least two of the best matching support images. This choice ensures that occlusions can be handled as only those images contribute to the cost function where the pixel under consideration is likely to be visible in. This procedure is similar to the one in [SZS03] where for a single baseline multi-view image set only the "best half sequence" in terms of intensity difference is used for matching. However, the cost function in (3.27) also can be used with multi-baseline image sets as evaluated in the results section though no selection with respect to the pose of the support views in $\boldsymbol{S}_v$ is performed.

The scanner cost incorporates the laser scanner depth information into the data cost. Due to the higher temporal resolution of the laser scans, the pose of the scans has to be interpolated from the device calibration. Figure 3.18 shows the point cloud obtained by using cubic splines to find the intermediate pose of the acquisition device from the sparse extrinsic calibration obtained from the camera images. The 300 shot sequence is used in this example and the position as well as the rotation of the acquisition device are determined using the global refinement passes during pose estimation. To ensure orthonormal rotation matrices during interpolation, the rotation representation is converted to unit quaternions.

After interpolation, the scanner data is projected into the current image using a Z-buffer to handle occlusions (Figure 3.22 in the top right shows an example of such a sparse per pixel depth map). Now, for some pixels a measurement from the laser scanner is available that can be incorporated into the cost function. The scanner cost is defined as:

$$C_s\left(\boldsymbol{p}, d_p\right) = \text{minimum}\left(\left|d_s(\boldsymbol{p}) - d_p\right|, d_{s,max}\right) - d_{s,max}. \tag{3.28}$$

This is a truncated linear model and defines a distance measure between the actively measured depth quantized and associated with depth label $d_s(\boldsymbol{p})$ and the depth associated with the label d_p. Note that C_s is negative and has its minimum at d_p=$d_s(\boldsymbol{p})$. As active depth measurements increase the probability of assigning depth label $d_s(\boldsymbol{p})$ to pixel $\boldsymbol{p}$, the data cost is decreased at and around this point. The parameter $d_{s,max}$ limits the impact of the scanner data on the data cost to avoid too much influence of a mismatch.

The feature point cost is generated from the sparse but very reliable depth information obtained during pose estimation via the 3D features. The cost function is set up analogous to the scanner cost function with a slightly different parameterization accounting for the more reliable depth information:

$$C_f\left(\boldsymbol{p}, d_p\right) = \text{minimum}\left(\left|d_f(\boldsymbol{p}) - d_p\right|, d_{f,max}\right) - d_{f,max} \tag{3.29}$$

The disparity cost function

The second term of Equation (3.25) defines the penalty for discontinuities and can also be referred to as the smoothness term. V denotes the cost for assigning depth labels to neigh-

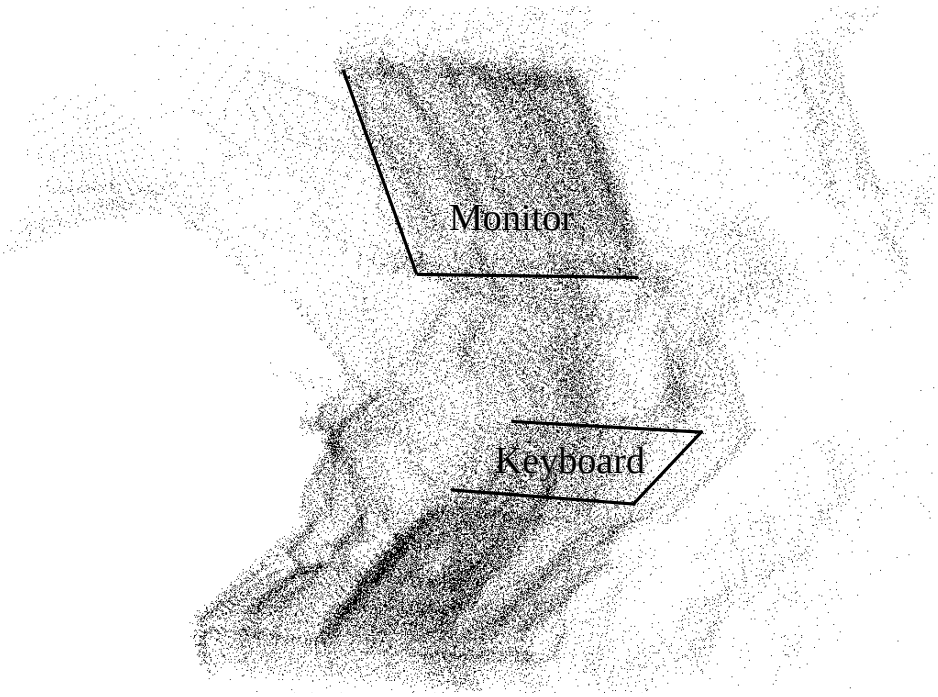

Figure 3.18: *Registered point cloud obtained from the depth samples of the laser scanner in the 300 shot sequence. The monitor and keyboard are sketched for clearer presentation.*

boring pixels:

$$V\left(\boldsymbol{p},\boldsymbol{q},d_p,d_q\right)=\min\left(\lambda_V(\boldsymbol{p},\boldsymbol{q})\cdot\left|d_p-d_q\right|,d_{V,max}\right) \tag{3.30}$$

Again, a truncated linear model is used as a distance measure between the depth for neighboring pixels d_p and d_q with $d_{V,max}$ as the maximum cost for a depth discontinuity. Additionally, color segmentation is performed by incorporating $\lambda_V(\boldsymbol{p},\boldsymbol{q})$ as a regularization of the smoothness dependent on the difference in normalized intensity of neighboring pixels:

$$\lambda_V(\boldsymbol{p},\boldsymbol{q})=a_V\cdot\left(1-\left|I(p)-I(q)\right|\right) \tag{3.31}$$

Here, a_V defines the weight of the color segmentation. This term causes discontinuities to be more probable at color segment boundaries.

3.5.2 Multi-view depth estimation for TRIVIS

To obtain dense per pixel depth maps for every single camera image with the depth estimation and fusion procedure introduced in the former sections, the image set $\boldsymbol{S}_I$ used for multi-view stereo reconstruction and the mapping from depth (-Z coordinate in the local camera coordinates of the image under consideration) to depth labels d_p have to be defined.

Reference view selection

To preserve local properties of the geometric model including the virtual depth of, e.g., specularities, a small subset $\boldsymbol{S}_I$ of N_S support images is selected from the whole set of images.

To simplify 3D reconstruction, the best images are chosen according to the proximity of their camera center and viewing direction to the image for which the dense depth map is to be generated. The selection procedure performs the following steps:

- Determine the mean depth of the scene for the image under consideration from the visible 3D features that have been obtained during pose estimation.
- Rank all input images according to
  - the spatial distance between their center of projection and the optical axis of the current view (ascending order with minimum distance of 4 mm).
  - the angular distance between their projection axes and the viewing direction of the current view (ascending order).
  - the relative coverage of the intersection of their viewing frustums with the image plane of the current view projected onto a plane at the mean distance of the scene (descending order).
- Average the rank of the cameras and pick the first camera and add it to $\boldsymbol{S}_I$.
- Rerank the cameras to ensure that the next best cameras' centers of projection are not all on the same side (and to ensure epipole consistency [BBM$^+$01]) of the current camera's viewpoint and repeatedly add some of the best cameras to $\boldsymbol{S}_I$.
- For some few cameras just rank the coverage, but during calculation of the coverage, do not consider the area that is already covered by other support cameras to make sure that every part of the whole current view is covered by at least one of the support images at the plane at the mean scene depth.

Figures 3.19 and 3.20 show the result of the above procedure for N_S=5 and where the last two added cameras are only ranked according to their coverage. In the X-Z plane as well as in the X-Y plane, the chosen camera centers are located around the current center of projection. The first cameras are chosen near the current camera. Cameras four and five are a bit farther away, but ensure that all pixels in the current view can be matched. The viewing directions are similar, and, cameras approximately lying at the same distance to the scene are chosen which ensures approximately the same resolution in the supporting images.

Depth label selection

The depth label selection is related to the joint image-geometry space discussed in Section 2.3.2 on page 14. While the minimum and maximum depth of the scene can be approximated from the 3D features obtained during the pose estimation steps, the number of depth layers needed for aliasing free view reconstruction (assuming perfect depth reconstruction and Lambertian surfaces as well as an occlusion free scene) can be calculated from Equations (2.3) and (2.7) on pages 11 and 14, respectively. As an example, with a minimum distance of scene points from the cameras of z_{min}≈0.5m, a maximum distance of z_{max}≈4m (from the 900 shot sequence), a focal length of f_c=1800px (from the camera calibration), and an approximated maximum distance between the camera centers of approximately ΔX_{max}≈6cm (compare to Figure 3.19), the minimum number of depth layers is set to N_D=100. The distribution of the depth layers is then chosen according to Equation (2.7).

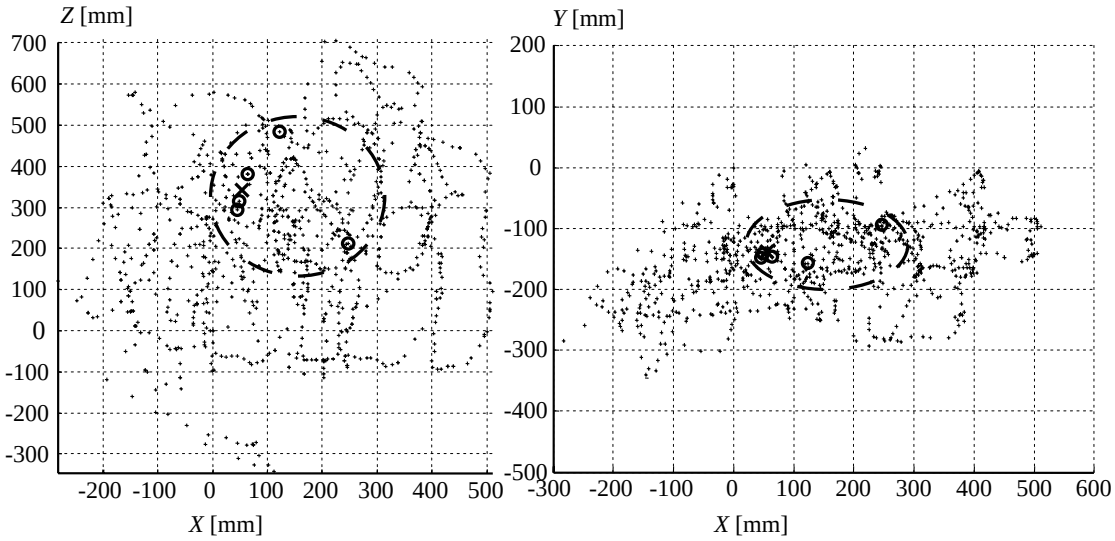

Figure 3.19: *Example for the view selection for multi-view stereo reconstruction. The dots denote camera centers for the 300 shot sequence. The big cross denotes the camera center for the image under consideration while the small circles mark the support images. The big dashed circle shows the local support of S_I.*

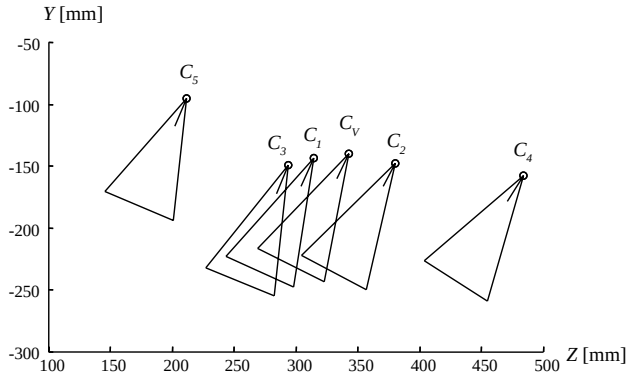

Figure 3.20: *The spatial distribution of the support views for stereo reconstruction. The image under consideration is C_V with N_S=5. The center of projection, the viewing direction, and the viewing frustums are indicated.*

Fusion results

The introduced sensor data fusion algorithm for dense depth estimation is tested on a line light field to demonstrate the performance of sensor data fusion as described in the former sections (only images from one camera and measurements from the laser scanner are used). The image sequence consists of 61 frames evenly spaced on a line and facing perpendicular to this line of length 45cm. Frame 31 (the middle image) is chosen to be the current frame. Figure 3.21 (top) shows the current image while in the left and right of the figure the depth solely obtained from the laser scanner data and the depth obtained from the images is

shown, respectively. The scanner range data is interpolated for clearer presentation. In the bottom of Figure 3.21 the fused depth map is shown. In untextured areas the laser scanner clearly has the major impact on the final result (the wall in the left). In areas with fine detail, the depth from the images has the greater impact on the final result (obstacles on the desk).

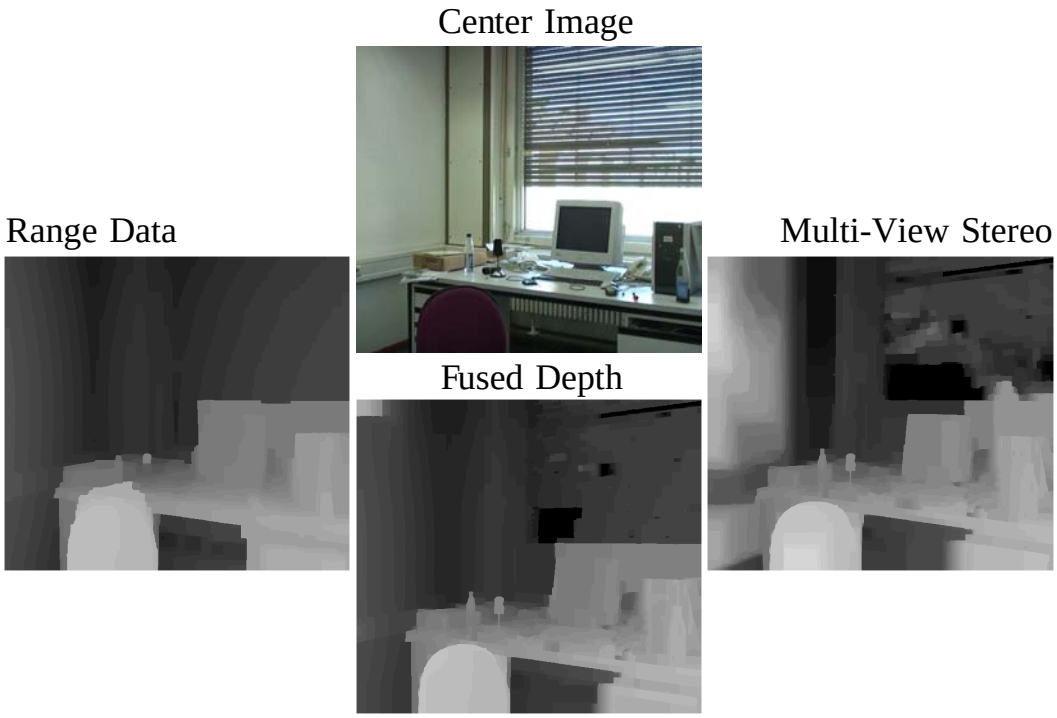

Figure 3.21: *Sensor data fusion for depth estimation. (top) Reference image. (left) Depth map interpolated from the point cloud obtained from the laser scanner. (right) Depth map generated from 10 images. (bottom) Results with fusion of the sensor data. Brighter pixels denote areas nearer to the camera. For black pixels no estimate is available.*

Table 3.7 gives the parameters that are used during the experiments with TRIVIS and have been determined manually. The intensity matching cost (3.27) is calculated using a plane sweep algorithm [Col96]. Minimizing the overall energy $E(\boldsymbol{F})$ corresponds to the maximum a posteriori estimation of the scene depth. An optimal labeling $\boldsymbol{F}$ and therefore an optimal depth map can be approximated using graph cut algorithms or Bayesian belief propagation algorithms (see Section 2.6 on page 21). However, results in this section are obtained by minimizing the overall cost function (2.6) using the algorithm described in [FH04] with modifications (incorporation of the Kullback-Leiber divergence (as suggested in [SZS03]) and a refinement by interpolating the depth labels according to the final local energy at a pixel location as described in [SS02]). For computational reasons, the images are subsampled to a resolution of 320×256 pixels during depth estimation. Sensor data fusion results for the data sets acquired with TRIVIS are shown in Figure 3.22. In the figure, the depth map obtained solely from images is shown in the upper left. Brighter pixels denote scene regions that are nearer to the camera. In the middle of the top row the depth obtained from the 3D features from the pose estimation steps is shown. It is very sparse while the depth obtained from the laser scanner measurements in the right of the top row is more dense. In the bottom row

on the left, the fused depth map is shown that clearly is more accurate than the three depth maps in the top row. The middle image in the bottom row shows the image that was subject to dense per pixel depth estimation while the right image shows the spatially nearest image in the image set $\boldsymbol{S_I}$. Note that the glass is not modeled properly by the laser scanner and the 3D feature data, but jointly with the multi-view camera data. On the other hand, flat areas that do not have much texture are better modeled by the scanner data.

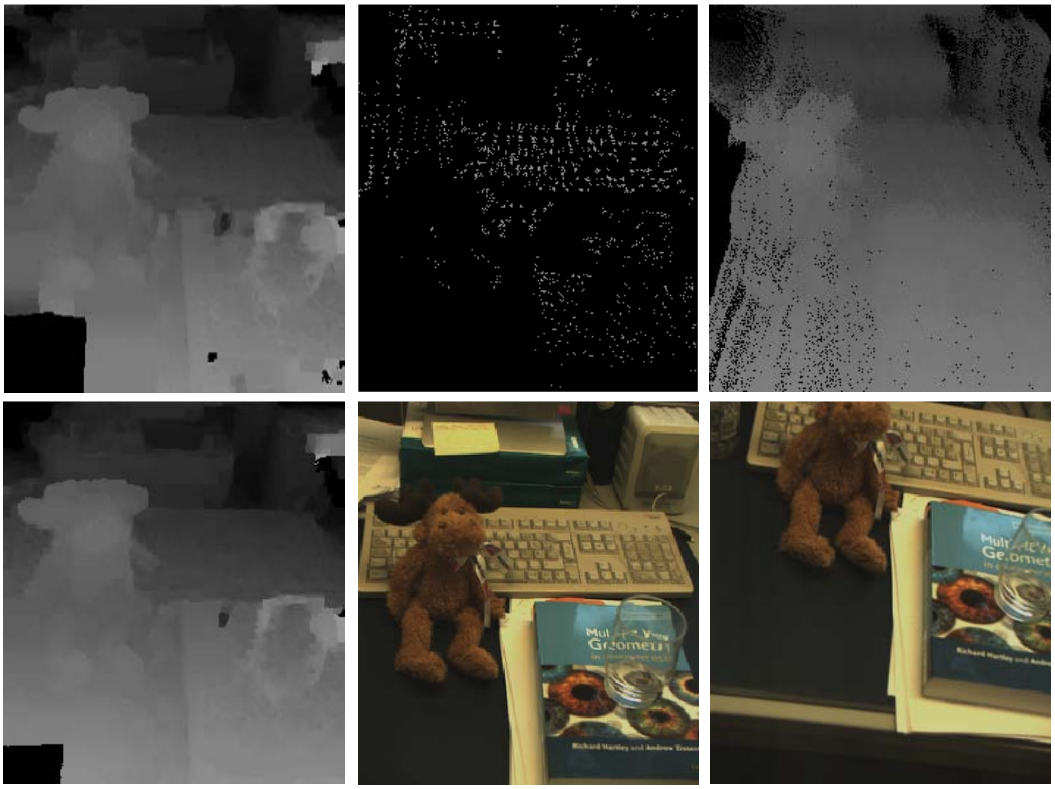

Figure 3.22: *Sensor data fusion results for one image of the 900 shot sequence. (top row) The depth obtained from the images solely, the depth obtained from the 3D features from the pose estimation steps, and the depth obtained from the laser scanner. (bottom row) The depth map after fusion of all depth cues, the image under consideration, and the spatially nearest support view. Brighter pixels denote areas nearer to the camera. For black pixels no estimate is available.*

3.6 Robust rendering

In this section a rendering algorithm is introduced that uses the local geometry models obtained in the former sections and the registered input images to perform photorealistic rendering of a static scene in real-time. Rendering is based on the idea to warp a set of support views selected from the input images of a light field into the desired virtual view and to blend them according to their proximity to the desired viewpoint (see Section 2.4.2 on page 18). To achieve real-time rendering, the warping and blending is performed on common dedicated graphics hardware which implies that the local scene models are present as meshes with texture information.

λ_s	weight of the scanner data	0.5
λ_f	weight of the 3D feature point data	0.7
c_{max}	maximum cost in the intensity matching function	100
$d_{f,max}$	maximum bonus for the 3D feature points	20
$d_{s,max}$	maximum bonus for the scanner measurements	20
$d_{V,max}$	maximum penalty for different depths in neighboring pixels	20
a_V	scaling of the disparity between neighboring pixels	0.5
N_S	number of support images to calculate the depth from	5

Table 3.7: *Summary of the parameters and settings for the depth fusion algorithm*

The per pixel depth maps retrieved in the former sections are simplified to a meshed representation using the software package "QSlim" [Gar06]. In the experiments, the final representation consists of the cameras' pose information, the input images themselves, 10000 3D vertices per image that are associated with (image) texture coordinates, and the information of connections between vertices to form a triangle mesh. Texture information is compressed and stored in JPEG format [IJ92] (high quality). This representation is accessed by a renderer which calculates the most appropriate subset of views to use for rendering just like the view selection introduced in Section 3.5.2 on page 64 and performs the warping and blending. The algorithm is robust to outliers in the depth estimation stage to some degree. This robustness is achieved by an on-the-fly outlier detection and removal algorithm. The rendering process is outlined as follows:

- Selection of the source images to use for rendering of a chosen virtual view
- Warping of the support images and their depth maps into the virtual view
- Per pixel blending weight calculation
- Outlier detection and removal
- View interpolation

3.6.1 Blending multiple local models

To blend the warped images, a standard blending algorithm is performed [BBM+01]. There, a local geometry is approximated from a common geometric model while the rendering system under consideration in this section warps the chosen input images into the virtual view using their local geometry representation. Then, blending is done according to the spatial distance between the centers of projection of the virtual view and the input images, the angular distance between corresponding viewing rays, the resolution of the input images with respect to the desired virtual view, and the visibility of the image content. Additionally, to avoid rubber band effects, a blending weight based on the face normal of the triangles is incorporated as suggested in [PSM03].

3.6.2 On the fly outlier removal

For scenes with a complex illumination, outliers during depth estimation are likely to occur. To handle such outliers, depth fusion could be performed, but in that case, local properties

are lost. Instead, during rendering, outliers are detected and removed. To achieve this, not only the input images are warped to the virtual view, also the depth maps are rendered. During blending, the per pixel depth in the warped support views are compared.

First the mean depth with respect to the virtual camera coordinates is calculated for all support views. Then, for every pixel in the virtual view, the weight for the pixel in the support view showing the largest depth deviation from the mean is set to zero if the absolute relative deviation from the mean depth is above a threshold Δr_{odr}. This procedure is repeated for all support views and pixels. Only those pixels with consistent depth remain and are blended to form the output color of the considered pixel in the virtual view.

3.6.3 Results

Figure 3.23 shows the impact of the outlier detection and removal procedure for the 900 shot sequence. 8 input images are selected, warped and used for blending. Outliers that occurred during depth estimation due to the complex illumination of the scene (natural lighting through the window and specular surfaces on the desk) are removed to some degree (Δr_{odr}=0.1, i.e., 10% deviation from the mean depth).

Figure 3.24 shows renderings using the high dynamic range imaging mode. Four image triplets are captured with the capturing device mounted on a tripod. Five exposures with different exposure times are triggered (2, 10, 30, 100, and 300ms). The top row in the figure shows the high dynamic range images (the [-4;1] relative irradiance value range linearly mapped to the [0;255] RGB space). The middle row shows two example renderings with a virtual exposure of 80ms. Additionally, there is an augmentation with a simple geometric primitive at the monitor where the rendered image is mapped to a polygon covering the entire visible screen after rendering. The viewing position of the left rendering is set in the middle of the line connecting source cameras two and three. The position of the virtual camera for the right rendering is set to be in the centroid of the input camera centers of projection which span an area of 20×20cm. The bottom row shows a detail of the left of the renderings in the middle row (without augmentation) at different virtual exposures (5, 50, 100, and 200ms). For short exposure times the content of the monitor is visible while for long exposure times details on the desk or behind the monitor are visible.

Figure 3.25 shows one final rendered view for the 900 shot sequence together with the camera viewing frustums of the input image data set, again, using the single exposure mode. The view selection procedure forces to evenly spread the views across the virtual view as indicated by the colored (grey) viewing frustums. The view is significantly far away from all input camera locations. Numerical results are shown in Table 3.8 and are obtained by subsequently removing one of the input images from the data set and rendering this view from the remaining images and local depth maps. The PSNR calculated by comparing the original view with the rendered one is averaged for a subset of 90% of the rendered virtual views with the largest PSNR. This selection is done to decrease the impact of rendered images at the borders of the covered viewing space.

The real-time performance of the rendering process is shown in Table 3.9. The bottleneck is the mesh and image loading from the hard disk and uploading to the texture memory of the graphics card. The test system is a Pentium IV dual core running at 2GHz and a NVIDIA GeForce 7900GS graphics card.

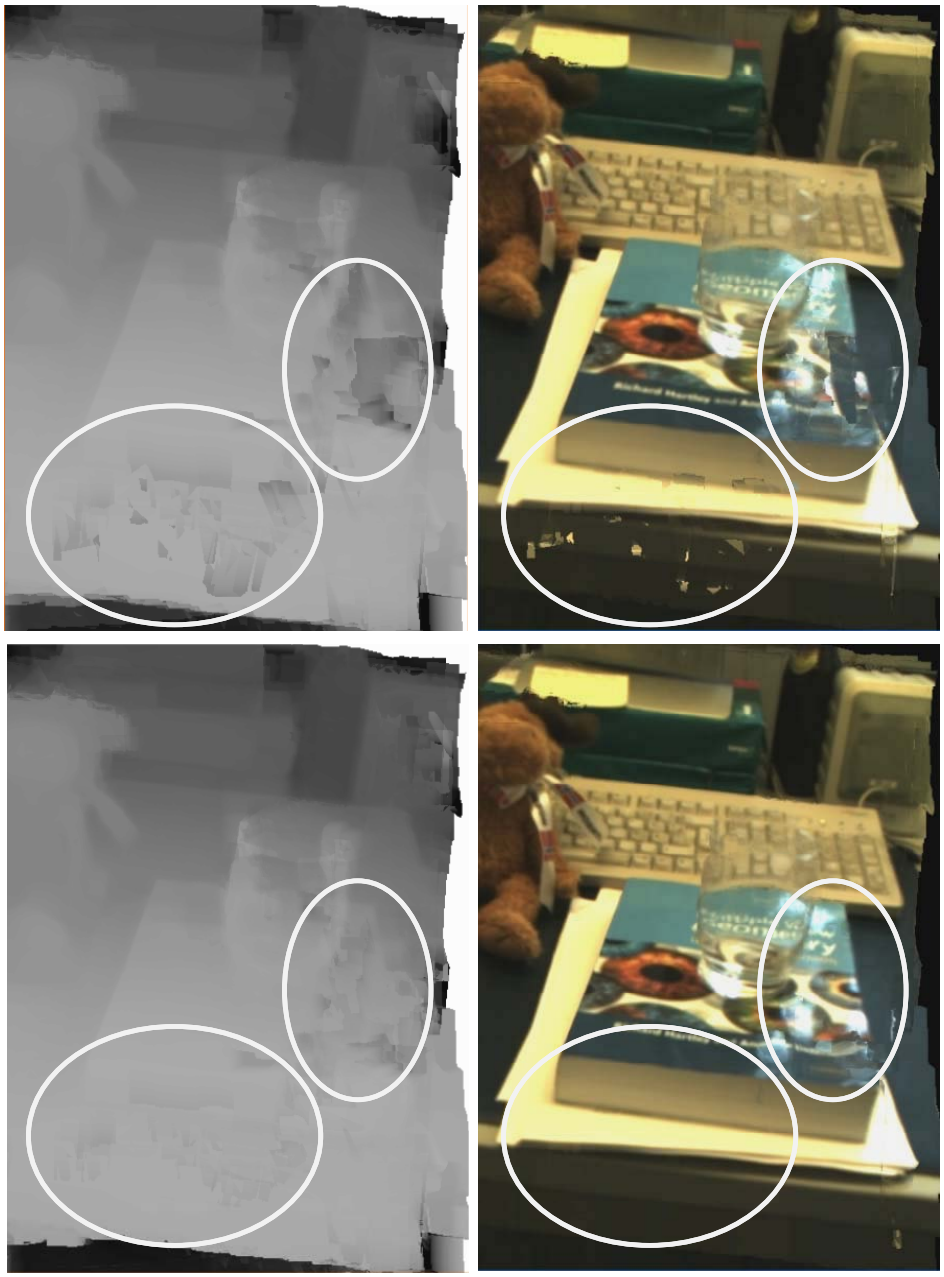

Figure 3.23: *The impact of the on-the-fly outlier removal algorithm. (top) Rendered depth map and final rendering result without outlier detection and removal. (bottom) Rendered depth map and final rendering result with outlier detection and removal. White ellipses indicate areas where the impact of the algorithm is significant.*

3.7 Discussion

The acquisition, calibration, pose estimation, scene geometry reconstruction, and rendering approaches discussed in this chapter are mostly based on existing techniques (e.g., [Tri98, TMHF99, BBM+01, ESK03, SMP05]). The full processing procedure from image acquisition

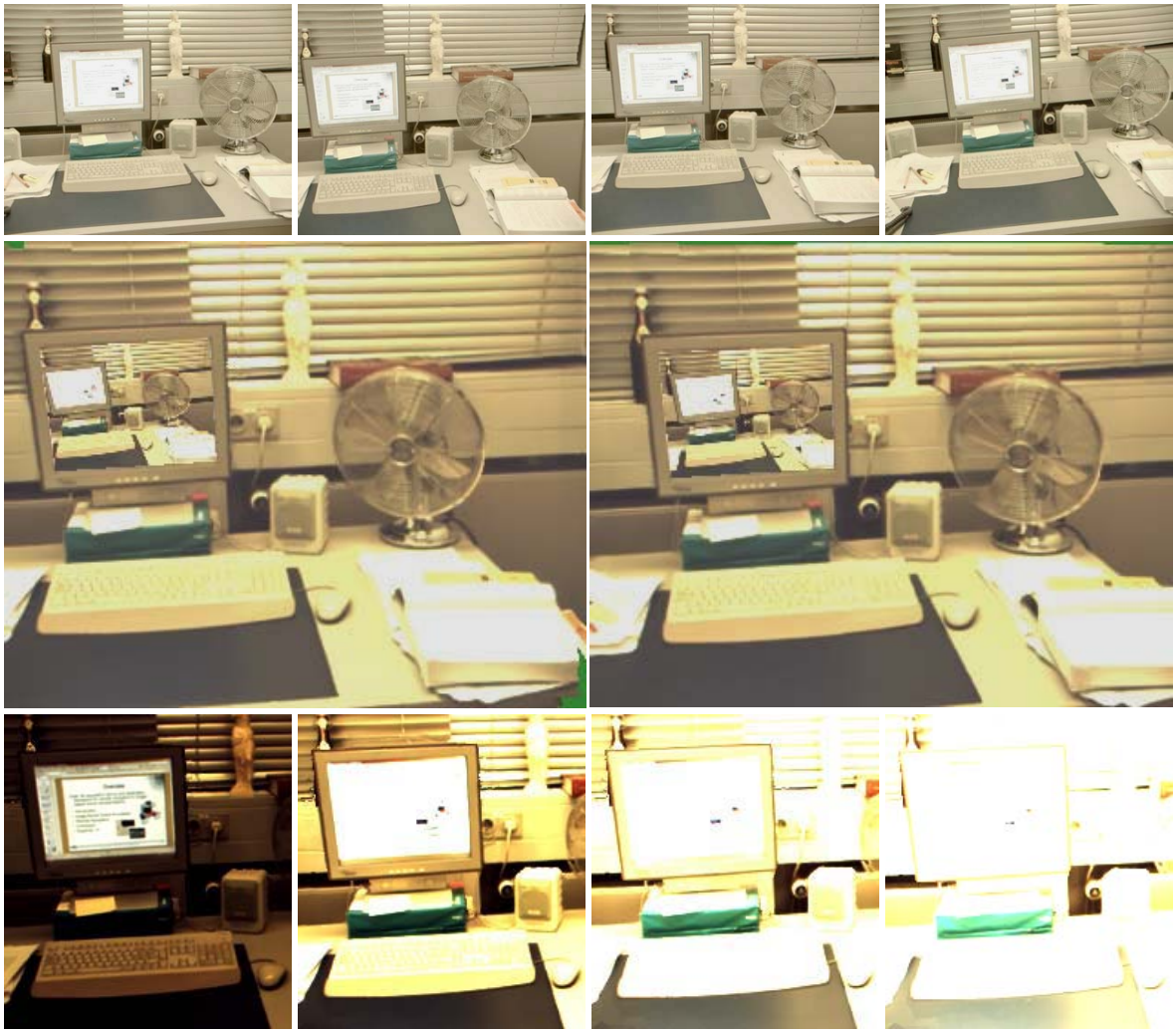

Figure 3.24: *High dynamic range rendering with TRIVIS (not using the laser scanner). (top row) The four source images that are taken from four registered high dynamic range shots. (middle row) Two renderings from positions half between the source images using a virtual exposure of 80ms. (bottom row) Rendering with different virtual exposures (5ms, 50ms, 100ms, and 200ms).*

	PSNR with outlier removal	PSNR without outlier removal
90 shot sequence	31.2dB	29.3dB
300 shot sequence	30.5dB	28.9dB

Table 3.8: *Numerical results for rendering with outlier detection and removal.*

mesh/image loading	view warping	outlier removal	blending
31ms	6ms	5ms	3ms

Table 3.9: *Real-time performance for rendering from the scene representation acquired with TRIVIS.*

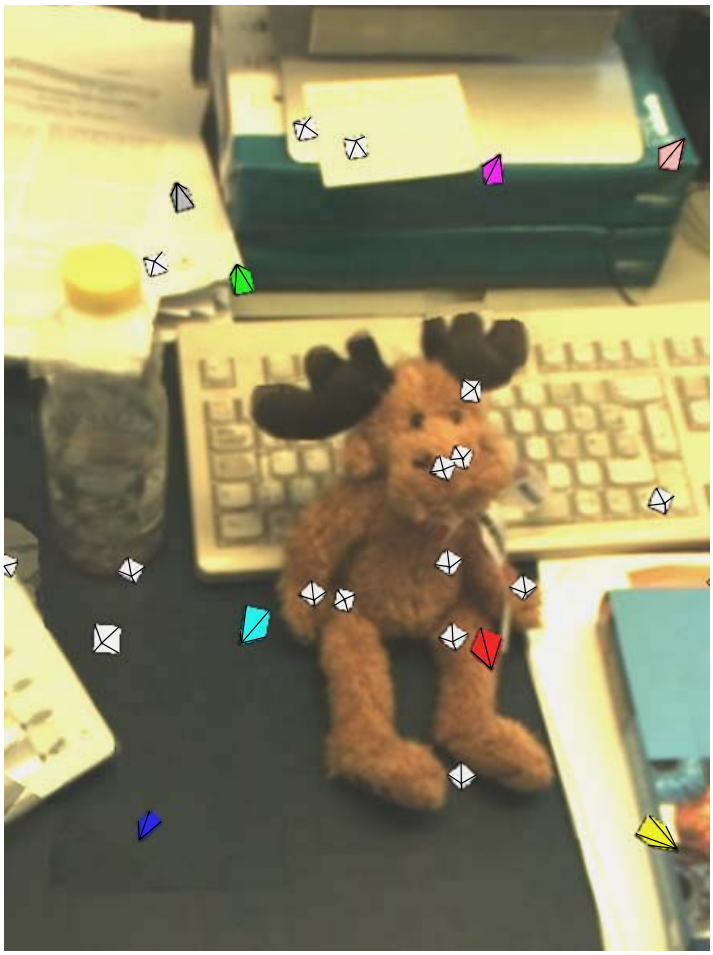

Figure 3.25: *A rendering using the proposed algorithm. The camera frustums that are used for rendering are indicated in color (gray) whereas the camera frustums of all images in the test set are indicated in white.*

to rendering has also been discussed in the literature (e.g., [HPDvG99, ESK05] where the former also uses a multi-sensor setup built of four cameras on a pole and the latter only uses one uncalibrated camera. Still, there are a couple of differences in the sensor setup as well as in the signal processing algorithms investigated in this chapter.

Jointly using a laser scanner together with multiple cameras on a hand-held device has not been investigated in detail for image-based rendering applications. Although efficient and elegant calibration algorithms for rigidly mounted multi-camera systems exist (e.g., [FKK04] assuming an uncalibrated multi-camera platform), the rematching algorithm based on direct pose estimation provides some advantages. First, fewer correspondences are needed per image for accurate pose estimation due to the rigid relation between the sensors [FKK04], further, mismatches are less likely due to guided matching. For the multi-camera systems found in the literature, rematching is not investigated, thus, long sequences can not be registered appropriately for image-based rendering. With respect to single camera systems, three images are acquired simultaneously with a well defined spacing by using TRIVIS, which allows a better coverage of the viewing space during acquisition. 3D reconstruction and rendering

of virtual views within the triangle spanned by the camera centers may be performed. This can already be done with a single exposure as accurate pose estimation is possible even for just a few acquired image triplets. Finally, pose estimation is more stable, as the registration of less textured scenes does not bear that high risk of loosing features during tracking.

The color calibration procedure is not new in its essence, but gives the system the flexibility to also capture scenes with complex and natural illumination such as outdoor sequences.

The 3D scene geometry reconstruction and sensor data fusion algorithm provides more reliable depth maps than reconstruction from a single camera or from multi-camera systems without a laser scanner. The cost function formulation can be easily adapted to weight the scanner data more or less, depending on the density of the scanner samples and their accuracy. This makes the fusion approach based on global energy minimization flexible also for other system configurations than that of TRIVIS. The scanner can not capture scenes with transparent objects like glass etc. very well. To capture scenes containing such objects, it is crucial to weight the intensity based part of the cost function more strongly than the scanner data or to significantly increase the sampling rate.

The rendering process is mainly performed using standard approaches. However, using multiple local models instead of a common geometric model and the outlier detection and removal algorithm allow for a temporally smooth rendering during motion of the virtual camera only using a small subset of the whole input image data. Holes due to disocclusions and occlusions in the warped image subset are handled with the incorporation of the surface normals into the per pixel weighting procedure.

The main difference to the systems most closely related to image-based rendering with TRIVIS is the use of the laser scanner. Further, local geometry models are computed densely with a global energy minimization approach. For rendering, multiple local models are used. The outlier detection and removal makes the rendering smooth during motion of the camera.

3.8 Summary

In this chapter image-based rendering from data acquired with a multi-sensor platform is investigated. The platform consists of three video cameras and a laser range finder that samples depth values on a plane. The device is to be moved manually in front of a scene to acquire images and laser scans that are registered in space and time to form an unstructured image-based scene representation. The registration process considers motion trajectories that crosses or comes near to itself to relate images that are spatially near. Per pixel depth maps for every single input image are obtained by sensor data fusion using area matching in the camera images, point features, and the laser range data. The rendering is performed in real-time and an outlier detection and removal algorithm is used to ensure time coherent photorealistic rendering.

The joint calibration algorithm provides a full metric calibration of the sensor device. Results show that the accuracy and stability of the discussed algorithm is comparable to common calibration methods only using images, but incorporates the extrinsic calibration of the laser scanner. The color calibration procedure allows for high dynamic range imaging as well as inter-camera color matching at an accuracy that is comparable to other multi-camera color calibration procedures. The image registration algorithm based on direct pose estimation provides reliable results even in cases where pose estimation with a single camera can

not be ensured over the whole sequence. The rematching algorithm allows for accurate registration of images that are temporally far apart but spatially near. Sensor data fusion results provide dense per image depth maps even with complex illumination with the support of the laser scanner. The rendering approach allows for on-the-fly per pixel depth outlier detection and removal in real-time using common graphics hardware. Visual as well as objective results are acceptable. Timing results show that high frame rates can be achieved to support immersive navigation for unstructured image-based scene representations.

The main contribution of this chapter is the joint calibration procedure for three cameras and a line laser scanner. Further, the rematching algorithm based on direct pose estimation which warps an added shot before rematching, allows us to track features even in the presence of heavy rotation and translation of the acquisition device. A further contribution is the global sensor data fusion algorithm for joint depth estimation based on belief propagation. Finally, the outlier detection and removal algorithm allows for rendering in the presence of outliers in the depth maps with a reduction of visual artifacts.

The hand-held multi-sensor device TRIVIS is designed to acquire unstructured image-based scene representations. A scene representation that consists of images and precalculated per pixel depth maps is used for virtual view generation. The geometric modeling procedure is prone to ambiguities as they might occur with untextured object surfaces, transparent objects, etc. Though this allows to reduce the number of required input images, a high density of the input image sequence might be preferable for certain applications to short cut the modeling procedure. The next chapters considers such densely sampled scene representations and the compression for interactive streaming.

4 RDTC optimized compression - A theoretical analysis

In this chapter theoretical aspects of the compression and interactive streaming of densely sampled and structured image-based scene representations are investigated. The input images are assumed to be encoded using disparity compensated prediction. As discussed in Section 2.8 on page 26 the degree of inter-frame dependencies exploited during encoding has an impact on the transmission and decoding time and at the same time delimits the (storage) rate-distortion trade-off that can be achieved for interactive streaming applications. To adapt to scenario specific properties like the available channel bit rate and computational power of the client device, the rate-distortion optimization approach that is commonly used for video and light field compression is extended to a trade-off between the storage rate (R), distortion (D), transmission data rate (T), and decoding complexity (C). A theoretical framework for RDTC optimized compression for interactive streaming is presented.

In Section 4.1 the considered streaming system is introduced and system measures and parameters are defined. Theoretical models for the decoding complexity of compressed image-based scene representations are given in Section 4.2. The rate-distortion model used is reviewed in Section 4.3. The system measures are incorporated into one RDTC model and the impact of client side caching on the system performance is evaluated in Section 4.4. In Sections 4.5 and 4.6, extensions to the framework as well as results are discussed. Section 4.7 gives a summary of this chapter.

4.1 System overview

Downloading of image-based scene representations over the Internet is infeasible in most cases as the number of images needed is often tremendous (see Section 2.3 on page 11). Rendering and encoding a chosen virtual view at the server using common still image or video compression is computationally too complex, especially in a multi-user scenario. Therefore, transmitting only the precoded image data that is actually needed for rendering a virtual view at the client side allows us to start a remote session instantly. The reconstruction quality, the system response time, and the frame rate are the most important measures that have an impact on the subjective feeling of realism of an interactive remote browsing application using image-based scene representations. Also taking the overall storage size into account, this motivates to develop special mechanisms for encoding and online operation to optimally provide the user with the data that is needed.

In this section the considered encoding and streaming system along with its properties is described in detail. An analysis of the properties of the input data that the considered system expects is given, and a description of the encoding process as well as the end-to-end signal flow during a remote navigation session is discussed. The system measures and encoding parameters used in the theoretical models are defined. As all theoretical models

are compared to experimental results during the following sections, a brief description of the error measures and the evaluation methodology is given in Appendices A.4 and A.5 on pages 162 and 163, respectively.

4.1.1 Compression using hybrid video coding concepts

Most image-based rendering systems use image sequences that have been captured using a calibrated video camera or camera arrays as the input to the image analysis and view synthesis steps. The minimum spacing of images within a real scene to avoid aliasing artifacts during view reconstruction has been discussed in Section 2.3 on page 11. In general, one can interpret spatially distributed camera positions in a static scene as motion trajectories of a single camera capturing a video sequence. In the following it is assumed that the input to the considered image-based streaming and rendering system is a calibrated RGB video sequence of a static scene or can be split into several of such sequences by partitioning the input image data. As the scene representation is also assumed to be critically sampled, neighboring images are very similar. Inter-image compression can exploit these similarities very efficiently. In common video coding standards [ITJ94, ITU00, Joi03] hybrid coders are used that can jointly exploit intra-image and inter-image redundancy. Figure 4.1 shows the block diagram of such a hybrid video coder [WOZ02].

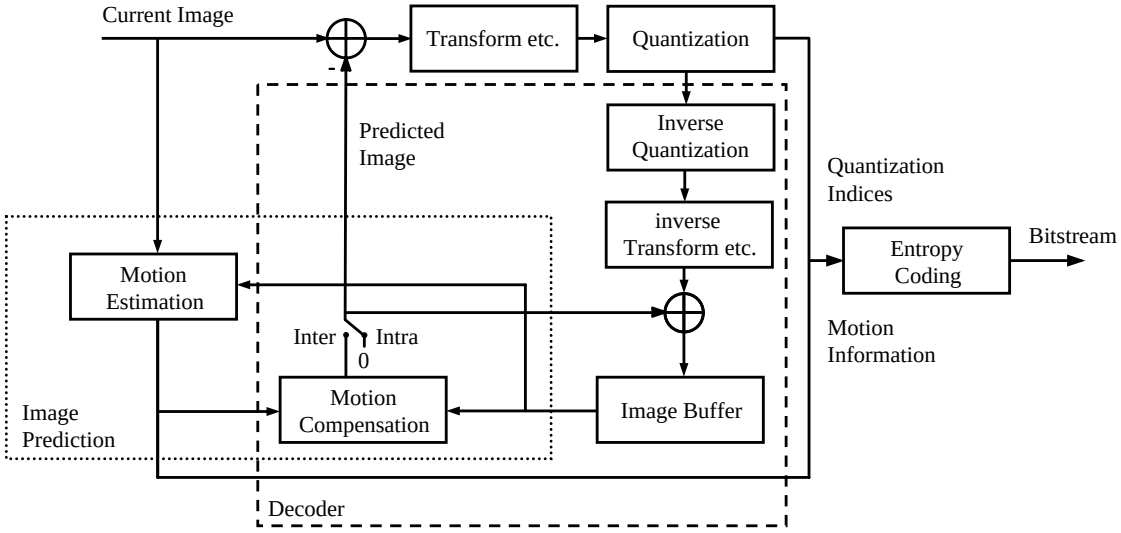

Figure 4.1: *The block diagram of a hybrid video coder*

Basically, differential encoding is performed. The current image is predicted by copying pixel blocks from previously encoded and decoded images in the image buffer. The relative location of the pixel blocks that are copied is estimated for every pixel block in the current image by minimizing an error measure (e.g., sum of absolute differences between the block in the current and one in the previous image) on a number of possible motion displacements. The predicted signal is subtracted from the current image to produce the residual error. This residual error is then further encoded, e.g., using a transform coding technique like the Discrete Cosine Transform (DCT) on pixel blocks to exploit spatial redundancy by

energy compaction. The transform coefficients are quantized to reduce the bitrate at the cost of lower reconstruction quality. Finally, the quantized transform coefficients and the motion displacement vectors are entropy coded. By setting the prediction signal to "0" (or to any other signal that is assumed to be known a priori by the decoder), independent encoding of images can be achieved. As the images that are used to estimate the prediction signal are decoded images rather than the original ones, the encoder and decoder both use identical prediction signals (closed loop prediction).

A variety of different algorithms for each of the building blocks has been proposed. Fractional-pel motion compensation as evaluated in [Gir93], bidirectional prediction [PAHL90], and multi hypothesis prediction [Gir00] among others have been presented to improve the reduction of temporal redundancy. Source models can be used (e.g., [SWG99]) and different transform coding schemes are possible in principle. Varying block sizes adaptively chosen with respect to the rate-distortion performance are used (e.g., [JVT03]), among a variety of other improvements.

The system considered in this chapter uses basic hybrid video coding concepts to keep the decoding complexity at a minimum. Offline compression on groups of pictures (GOPs) of size N images, each of size N_xxN_y pixels is assumed to be performed. Subscripts x and y denote horizontal and vertical image coordinates, respectively. Figure 4.2 illustrates the coding structure and example block modes. Consecutive frames are assumed to be encoded using

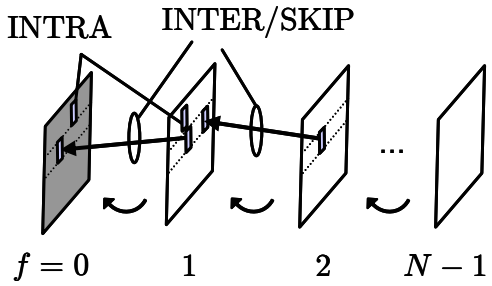

Figure 4.2: *GOP of N frames and example block modes. Frame f=0 is encoded independently (INTRA). Arrows denote dependencies among images and blocks.*

disparity compensated prediction on non-overlapping B_xxB_y pixel blocks on a regular grid. The motion displacement vector $\Delta\boldsymbol{x}$ (in pixels) of a pixel block in consecutive frames is calculated by minimizing the mean squared error (see Section A.4 on page 162) between the luminance intensities of the pixels in the current and reference blocks. As a static scene is assumed to be captured and due to the epipolar constraint (e.g., [LH81]), the 2D motion vector of a pixel block can be described by a scalar displacement Δd using a mapping $\boldsymbol{D}(x,y,\Delta d)$ which can be determined from the frame calibration (compare to Section 2.7.3 on page 24). It is assumed that the dominant motion within a GOP is in the horizontal direction. This assumption would force to perform partitioning and resampling of image sequences that are arbitrarily aligned in space. Though this reduces the overall system performance, such a representation is still valid as input. Then, the motion displacement vector $\Delta\boldsymbol{x}$ of a pixel block at position (x,y) can be written as:

$$\Delta\boldsymbol{x} = \boldsymbol{D}\left(x, y, \Delta d\right) = \begin{bmatrix} \Delta d \\ 0 \end{bmatrix}. \tag{4.1}$$

Consequently, only horizontal disparity compensation is performed. This simplification significantly reduces the complexity of disparity estimation for encoding and disparity compensation for decoding and also reduces the bitrate for disparity vector encoding and transmission. The assumption that an approximately rectified sequence is used also reduces the overall image quality when resampling has to be performed for rectification. Nevertheless, the principles discussed in the following sections also apply for approximately rectified sequences like those captured for most structured scene representations without the need for resampling. Issues regarding GOP structures with camera placements on a 2D grid, hierarchical GOP structures, as well as GOPs with branches in the prediction chain are discussed in Section 4.5 on page 101.

Pixel blocks can be encoded in INTRA and INTER/SKIP modes. Pixel blocks encoded in INTRA mode are encoded independently from any other intensity values in the reference data. The specific encoding scheme is not important for the theoretical analysis in this chapter as long as it is significantly more complex than simple pixel copying. For the INTER block mode the residual error after disparity compensated prediction is encoded using the INTRA encoding scheme. Blocks encoded in SKIP mode are predicted using disparity compensation while the residual error is not encoded.

4.1.2 Interactive streaming

The streaming system considered in this chapter is depicted in Figure 4.3. The encoded bitstream is stored at the server. During a remote streaming session, a client application requests the compressed representation of a virtual view from the server. The pixel blocks containing relevant pixels for rendering are transmitted and corresponding reference blocks that are not present in the client's cache are transmitted as well until all reference blocks can be decoded at the client side. The smallest decodable unit in the considered system is a single pixel block, no matter if the requested image part is only a fraction of a pixel block like one pixel (as it might happen for a structured or unstructured light field representation [TG03]) or a pixel column (as it might happen for concentric mosaics [SH99]). A synchronization mechanism ensures that a cache index table (P-Index) provides the server with the knowledge of the client's cache state prior to any transmission. Once compressed data is received by the client, it is decoded, the cache is updated, and the virtual view is interpolated from the decoded pixel data and displayed by the renderer. The transmission data rate might be limited, and therefore a transmission delay is introduced. Also the decoding speed of the client device is limited what might introduce a delay for decoding the streamed data.

4.1.3 System measures and parameters

Traditionally, image and video compression has been studied within the rate-distortion theory framework. In the context of interactive streaming of image-based scene representations, the random access requirement puts additional constraints on the decoding complexity and transmission data rate.

RDTC system measures. The system measures used in this work to capture the properties of a compressed scene description are defined as follows:

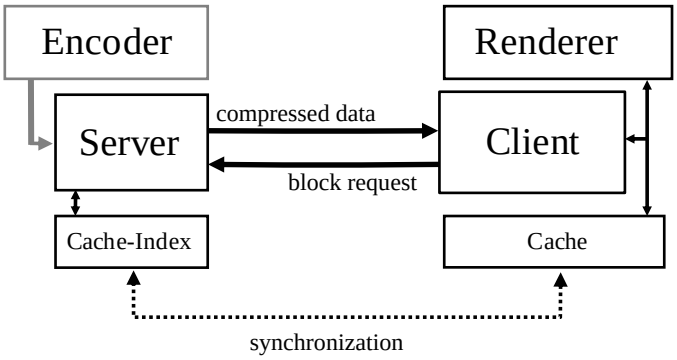

Figure 4.3: *The block diagram of the streaming system.*

1. The **Rate (*R*)** measured in bit per pixel (bpp) is the mean number of bits required to store the compressed representation of a pixel's RGB intensity values at the server.
2. The **Distortion (*D*)** is defined as the mean of squared differences (Section A.4 on page 162) of the luminance values of one reference block or image, measured using the difference between original intensity values and intensity values reconstructed from the compressed bitstream.
3. The **Transmission data rate (*T*)** is defined as the mean number of bits that have to be signaled to completely decode a pixel (bpp). The transmission data rate T can be significantly larger than R as dependencies might have to be resolved.
4. The **Decoding complexity** (***C***) measured in pixel per pixel (ppp) is the mean number of pixels that have to be decoded to reconstruct a pixel at the client. The decoding complexity C can be significantly larger than 1ppp as dependencies might have to be resolved.

The differentiation of these measures is chosen as they allow for a direct mapping from encoding parameters to metrics in a real system (like file size, PSNR, bandwidth and clock rates).

The mean response time. The overall performance of the streaming system is measured by the user perceived delay. As it is assumed that the decoder instantly starts decoding after receiving the first bit from the compressed virtual view, the mean response time t_d to user input is:

$$t_d = V \cdot \max\left(\frac{T}{r_{max}}, C \cdot t_C\right) + t_{rtd} \tag{4.2}$$

where V is the mean number of reference pixels required to display a virtual view (i.e., the number of pixels actually needed for view interpolation. This number might vary depending on the chosen viewpoint). r_{max} is the maximum available transmission bitrate and t_C is the mean time to decode a pixel encoded in INTRA mode at the client side. For the analysis introduced in this chapter, the fixed round trip delay t_{rtd} is ignored. Rather than specifying a deadline for the presentation of a virtual view as it can be found in the literature and which requires online scheduling, the goal is to optimally compress a scene representation in the

(storage) rate-distortion sense so that (4.2) holds for a desired resolution and mean response time t_d given scenario specific constraints r_{max} and t_C.

The INTRA and SKIP ratios. The analysis presented in the following requires some additional definitions of parameters used to specify the encoding process. As blocks encoded in INTRA mode - in contrast to INTER/SKIP blocks - can be decoded without any dependencies to be resolved, the INTRA ratio α' is an important measure for a random access analysis of a compressed image sequence. α' is defined as the ratio of independently encoded blocks in a GOP:

$$\alpha' = \frac{S_{INTRA} \cdot B_x \cdot B_y}{N \cdot N_x \cdot N_y} \tag{4.3}$$

where S_{INTRA} is the number of INTRA encoded blocks in a GOP. Due to the non-uniform distribution of the INTRA mode (the first frame of each GOP is encoded entirely in INTRA), the INTRA ratio α for all frames except the first INTRA frame is defined as:

$$\alpha = \frac{\alpha' \cdot N - 1}{N - 1}. \tag{4.4}$$

Correspondingly, the SKIP ratio β is defined as

$$\beta = \frac{S_{SKIP}}{S_{INTER} + S_{SKIP}} \tag{4.5}$$

where S_{INTER} and S_{SKIP} are the number of INTER mode and SKIP mode encoded blocks in a GOP, respectively.

The intrinsic pixel block size. The intrinsic pixel block size B_e is the block size that is used internally due to interpolation when performing disparity compensation in sub-pel accuracy. It is defined as

$$B_e = 2^{(s-1)} \cdot B_x \tag{4.6}$$

where s is the fractional-pel disparity compensation accuracy factor (1 for full-pel accuracy, 2 for half-pel accuracy, 3 for quarter-pel accuracy,...).

The single reference block ratio. A parameter to describe the disparity properties of a GOP is introduced as the single reference block ratio b. b is the ratio of blocks that have a reference block in a neighboring frame that falls exactly on the block grid. Correspondingly, 1-b is the ratio of pixel blocks having two reference blocks in the reference frame. Figure 4.4 illustrates the meaning of b. Assume the block containing the scene object in the current frame on the left is to be decoded. The disparity of the scene object Δd is not a multiple of B_x (the disparity is denoted by the bold arrow). This forces the system to decode the two grey blocks in the previous frame. In the right of Figure 4.4 the single reference block case is illustrated. Again, the block containing the scene object in the current frame is to be decoded. As the disparity Δd now is assumed to be a multiple of B_x, only one block in the previous frame serves as a reference.

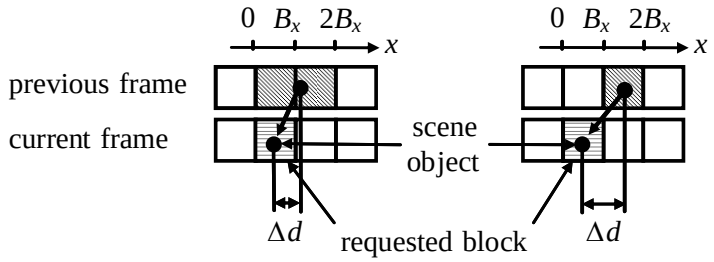

Figure 4.4: *Illustration of the double reference case (left) and the single reference case (right). The single reference block ratio b is a signal dependent parameter.*

The single reference block ratio b is signal dependent and defined with respect to a specific realization of a disparity field within a GOP:

$$b = \sum_{k=-\infty}^{\infty} p_{\Delta d}\,(k \cdot B_x). \tag{4.7}$$

Here, $p_{\Delta d}$ is the probability mass function of the disparity Δd. Assuming a uniform distribution of Δd, b can be approximated by

$$b \approx \frac{1}{B_e}. \tag{4.8}$$

The single reference block ratio has a major impact on the decoding complexity as shown in Appendix A.8 on page 166 with respect to the analysis in Section 4.2.3 on page 90 of this chapter. Typically, for densely sampled input images, b can be approximated using (4.8).

The quantization parameter. In practical experiments throughout this chapter the quantization parameter q is used to control the rate-distortion trade-off in a H.263 [ITU00] manner (deadzone quantization on visually weighted transform coefficients). Similarly, the noise parameter θ is used to produce rate-distortion curves in the theoretical analysis. It is assumed that q can be expressed as a function of θ.

Summary of measures and parameters. Table 4.1 summarizes the encoding parameters and system measures that are mapped by the theoretical analysis in subsequent sections.

4.2 The decoding complexity

In this section the decoding complexity as defined in the previous section will be analyzed in detail. A motivation from a practical point of view is given and decoding complexity models for simple single random block access patterns are derived. Models for more complex virtual view access patterns as can be observed in real systems follow.

Independent from the INTRA encoding scheme used, which can be based on transform coding or any other technique like vector quantization, the decoding complexity C for decoding one pixel block from the compressed image sequence is defined as the number of

Encoding Parameters	
GOP size N	Number of frames per GOP
Frame size N_x, N_y	Height and width of a frame in pixels
Block size B_x (=B_y)	Pixel block width (height) in pixels
INTRA ratio α	Ratio of INTRA blocks for frames $f \in [1, N-1]$
SKIP ratio β	Ratio of SKIP blocks (among non-INTRA blocks)
Single block ratio b	Ratio of disparity values that are integer multiples of the block size
Quantization parameter $q(\theta)$	Deadzone quantizer step size
Disparity estimation accuracy s	Fractional pel accuracy: $2^{-(s-1)}$

System Measures	
Rate R [bpp]	Storage rate
Distortion D (MSE)	Reconstruction quality (reference data)
Decoding complexity C [ppp]	Mean number of pixels decoded per requested pixel
Transmission Rate T [bpp]	Mean number of bits transmitted per requested pixel

Table 4.1: *Summary of encoding parameters and system measures*

pixels that have to be decoded to reconstruct the RGB intensity values of a pixel completely (decoded pixel per requested pixel [ppp]). This definition is chosen because blocks in INTRA and INTER mode consume most of the decoding time while SKIP blocks only require copy operations. To support this statement, Table 4.2 shows the result of an experiment where the processor cycles needed for decoding different block modes at medium/high quality are counted for a roughly optimized straight forward implementation. Note that the INTRA/INTER decoding procedure is over 20 times more complex compared to the SKIP mode. Similar findings have been presented in [MG00a, VNP02], though not showing that significant differences.

Block Mode	Mean number of processor cycles per 8x8 block
INTRA	14000
INTER	15000
SKIP	600

Table 4.2: *Decoding complexity for different block modes*

The decoding complexity C is defined recursively and with respect to the block disparity Δd_B (unit: block) which is calculated from the disparity Δd (unit: pixel) at a position $\boldsymbol{p} = (x, y)$ in frame f:

$$\Delta d_B(\boldsymbol{p}, f) = \left\lfloor \frac{\Delta d(\boldsymbol{p}, f)}{B_x} \right\rfloor . \tag{4.9}$$

For the simple case of INTRA encoding, a block is independent from other reference images

and the decoding complexity C can be written as

$$C(\boldsymbol{p}, f) = 1 \text{ [ppp]}. \tag{4.10}$$

When decoding from INTER mode the residual error that has been encoded in INTRA mode has to be considered. Therefore, the residual decoding complexity is set to M=1 when encoding in INTER mode and M=0 in SKIP mode (though M can take on any value if the SKIP decoding procedure is considered more complex). For the case with a single reference block (compare to Section 4.1.3 on page 82), i.e., $\Delta d(\boldsymbol{p}, f)(\text{mod } B_x)$=0 ($\Delta d$ is a multiple of B_x), C can be written as

$$C(\boldsymbol{p}, f) = M + C\left(\left[\begin{array}{l} x + B_x \cdot \Delta d_B(\boldsymbol{p}, f) \\ y \end{array}\right], f-1\right) \tag{4.11}$$

and for the case that $\Delta d(\boldsymbol{p}, f)(\text{mod } B_x) \neq 0$:

$$\begin{aligned} C(\boldsymbol{p}, f) = \quad & M + C\left(\left[\begin{array}{l} x + B_x \cdot \Delta d_B(\boldsymbol{p}, f) \\ y \end{array}\right], f-1\right) \\ & + C\left(\left[\begin{array}{l} x + B_x \cdot (\Delta d_B(\boldsymbol{p}, f) + 1) \\ y \end{array}\right], f-1\right). \end{aligned} \tag{4.12}$$

For the case that a block is already decoded and present in the client cache, the decoding complexity vanishes:

$$C(\boldsymbol{p}, f) = 0 \text{ [ppp]}. \tag{4.13}$$

An example for a single block request is illustrated in Figure 4.5. In frame f a block (dashed square) is to be decoded. As it is encoded in INTER mode, dependent blocks have to be decoded. In this example, the disparity is an integer number $0 < \Delta d < B_x$. In the reference frame (f-1) two blocks "A" and "B" have to be decoded as Δd is not a multiple of B_x. These blocks in turn have dependencies to be resolved. For "A" the disparity relative to its reference frame f-2 is Δd=0. Only one reference block in f-2 has to be decoded for this block. A similar procedure is performed until in frame f-4 all blocks are encoded in INTRA mode. Now, the originally requested block can be decoded completely. Assuming an empty cache before the request, the decoding complexity is C=15 pixels per pixel in this example, assuming all pixels in the requested bock are used for rendering. Equations (4.10)-(4.13) can be evaluated by experiment to determine the complexity of a single random access experiment for a given compressed bitstream and access pattern. The mapping from C to processor cycles depends on the INTRA encoding scheme chosen and the target platform.

4.2.1 Decoding a single block without a cache

In this section a model for the decoding complexity for an IBR rendering system not implementing a cache and performing random access to single blocks is derived. Consider a sequence of images divided into groups of pictures of size N and compressed only using blocks which depend on two reference blocks in the previous frame. The worst case decoding complexity C_W is observed when decoding a block in a frame with maximum distance to the INTRA only encoded first frame of the GOP. C_W depends on the size of the GOP N and can be calculated as follows:

$$C_W(N) = \sum_{l=0}^{N-1} 2^l. \tag{4.14}$$

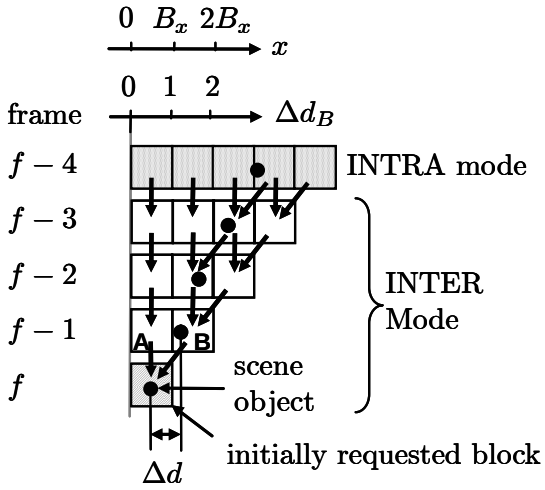

Figure 4.5: *Example decoding structure for a single block request. Bold arrows denote dependencies. The solid black circle represents a scene object. See text for a detailed explanation.*

This is the summation of all dependencies in all frames of the GOP plus the requested block itself. Now it is assumed that an arbitrary block from the bitstream has to be decoded, then, the distance between the requested frame and the independently encoded frame is not the fixed value N-1. The resulting mean decoding complexity C_{SNCI} (subscript SNCI stands for Single block access with No Cache and INTER only encoding) for access somewhere within the GOP without caching can be written as:

$$C_{SNCI}(N) = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{l=0}^{f} 2^l. \tag{4.15}$$

Now, a fraction α of blocks is allowed to be encoded in INTRA mode. The mean decoding complexity C_{SCNII} without caching but considering INTER and INTRA modes can be approximated as follows:

$$C_{SNCII}(N, \alpha) = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{l=0}^{f} 2^l \cdot (1-\alpha)^l. \tag{4.16}$$

Also, it has to be considered that a fraction of INTER blocks might have one reference block while the rest have two reference blocks in the previous frame. This is reflected by the single reference ratio b. The mean decoding complexity C_{SCNIII} can now be written as

$$C_{SNCIII}(N, \alpha, b) = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{l=0}^{f} \left((2-b) \cdot (1-\alpha)\right)^l. \tag{4.17}$$

The mean decoding complexity observed for blocks in the INTRA frame in the beginning of the GOP is:

$$C_{GSNC}(N, \alpha, b) = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \left((2-b) \cdot (1-\alpha)\right)^f. \tag{4.18}$$

Finally, when considering that a fraction of blocks can be encoded using SKIP mode, then the complexity of blocks in frames $f > 0$ have to be scaled according to the INTRA and SKIP ratio to get the final complexity for the case of a single block random access:

$$C_{SNC}(N, \alpha, b, \beta) = C_{GSNC} + (C_{SNCIII} - C_{GSNC}) \cdot (1 - \beta \cdot (1 - \alpha)) . \tag{4.19}$$

Figure 4.6 shows the comparison between a theoretical evaluation using (4.19) and experimental results for a GOP size of N=10 frames and different block sizes B_xxB_y. The experimental results in Figure 4.6 and in subsequent sections are obtained following the methodology described in Appendix A.5 on page 163.

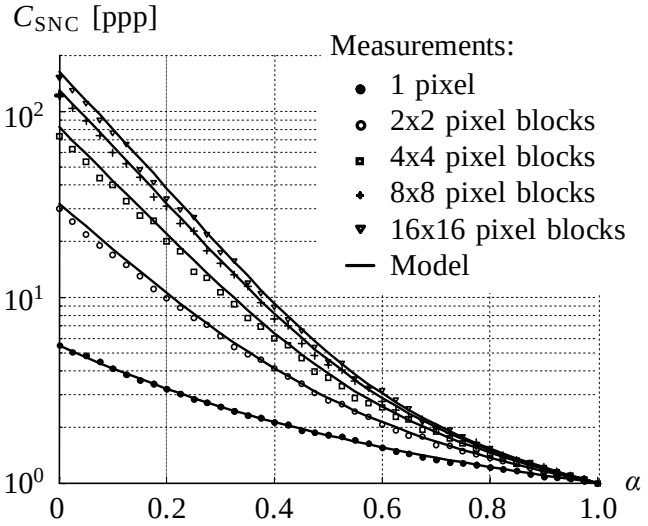

Figure 4.6: *The decoding complexity as a function of the INTRA ratio α for a GOP size of N=10 frames from (4.19) without caching and β=0.*

4.2.2 Decoding a single block with a pixel domain cache

When a sufficiently large pixel domain cache is used by the client, modeling the decoding complexity can be done using a statistical model. Figure 4.7 illustrates the statistical relationship between dependent blocks in neighboring frames. Assume block $(m, n) = (0, 0)$ in the right of Figure 4.7 is requested (i.e., $\Delta d_B = 0$ in $f = 0$ as shown in the left of Figure 4.7), then the probability that it has to be decoded is $a_{0,0} = 1$. Block $(0, 1)$ will not be decoded in this pass, so $a_{0,1} = 0$. The probability that block $(1, 0)$ will be decoded depends on α and the requesting block $(0, 0)$, i.e., $a_{1,0} = 1 \cdot (1 - \alpha)$ because only when $(0, 0)$ is in INTER/SKIP mode there are any references to resolve. Similarly $a_{1,1}$ is calculated, but, now block $(0, 0)$ and $(0, 1)$ might reference this block. Here, three cases can be identified:

1. $(0, 0)$ is INTER/SKIP and does not have a single reference; $(0, 1)$ has not to be decoded.
2. $(0, 1)$ is INTER/SKIP; $(0, 0)$ has not to be decoded.
3. both, $(0, 0)$ and $(0, 1)$ are being decoded and at least one of them is encoded in INTER/SKIP mode while $(0, 0)$ has not a single reference if $(0, 1)$ is not in INTER mode.

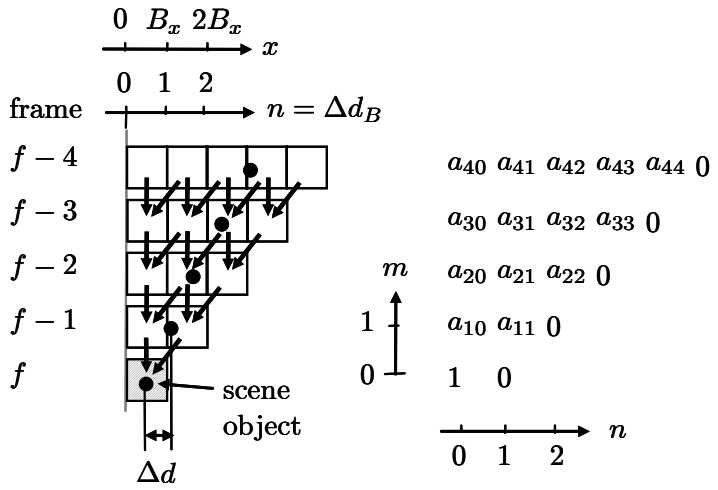

Figure 4.7: *Illustration of a statistical model for the decoding complexity in systems with pixel domain caching. Bold arrows indicate dependencies between blocks in neighboring frames. Block (f,n=0) on the left hand side is requested and decoded with a probability $a_{m,n}$ as denoted on the right hand side. m denotes the relative position to the requested block in the prediction past.*

Adding the corresponding expressions for those three cases gives the resulting probability for decoding the block $(m, n) = (1, 0)$ that is substituted using $j_{m,n}$ for later use:

$$\begin{aligned} j_{m,n} \equiv \quad & a_{m-1,n-1} \cdot (1-\alpha) \cdot (1-b) \cdot (1-a_{m-1,n}) \\ & + a_{m-1,n} \cdot (1-\alpha) \cdot (1-a_{m-1,n-1}) \\ & + a_{m-1,n-1} \cdot a_{m-1,n} \cdot \left(\left(1-\alpha^2\right) - \alpha \cdot b \cdot (1-\alpha) \right). \end{aligned} \tag{4.20}$$

With (4.20) the recursive expression for the probability that a block has to be decoded becomes:

$$a_{m,n} = \begin{cases} 0 & \text{if } n > m \\ 1 & \text{if } m, n = 0 \\ a_{m-1,n} \cdot (1-\alpha) & \text{if } m \neq 0; n = 0 \\ j_{m,n} & \text{else.} \end{cases} \tag{4.21}$$

Averaging over all possible random access points in a GOP and all blocks with $a_{m,n} > 0$ yields the decoding complexity C'_{SC} for a single block random access event for systems with a sufficiently large pixel domain cache implemented:

$$C'_{SC} = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{l=0}^{f} \sum_{t=0}^{l} a_{l,t}. \tag{4.22}$$

The decoding complexity only caused by blocks in the INTRA only frame of the GOP is:

$$C_{GSC} = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{t=0}^{f} a_{f,t}. \tag{4.23}$$

When adding the SKIP mode to the possible block modes, the decoding complexity can be calculated similar to (4.19). The decoding complexity C_{SC} for a single block random access

event with cache and INTRA/INTER/SKIP modes can be written as:

$$C_{SC}(N, \alpha, b, \beta) = C_{GSC} + \left(C'_{SC} - C_{GSC}\right) \cdot (1 - \beta \cdot (1 - \alpha)) \,. \tag{4.24}$$

Figure 4.8 shows the comparison between a theoretical evaluation using (4.24) and experimental results for a GOP size of N=10 frames and different SKIP rates β, the block size is B_x=8. Compared to the case without a cache (Figure 4.6) the decoding complexity has drastically decreased for low values of α. For β=0 the mean number of blocks that have to be decoded for a single block request is five times less when using a cache compared to the case without a cache. For large INTRA ratios (α>0.6) almost no reduction can be observed. The reduction increases when N is increased (not shown in the figure).

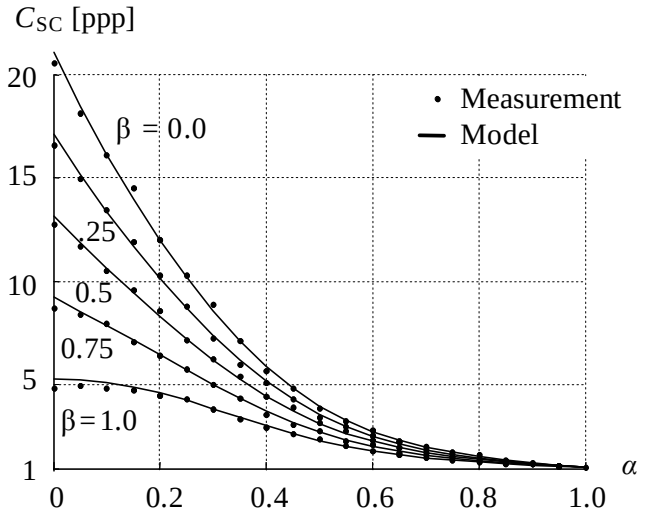

Figure 4.8: *The decoding complexity as a function of α and β for a GOP size of N=10 frames and a block size of B_x=8 using (4.24).*

4.2.3 Decoding virtual views (with cache)

Up to now only single blocks have been decoded. However, in image-based rendering systems, complete virtual views consist of image data from nearby frames and blocks which, again, share reference blocks. With a sufficiently large cache, the decoding complexity can be significantly reduced. Figure 4.9 shows typical access patterns to densely and regularly sampled image-based reference data. The x-axis denotes the frame number f and the y-axis denotes the block number d_B. Decoding one of the captured image block rows results in the access pattern denoted as "A". A very similar access pattern occurs for virtual views near reference camera positions. Access pattern "B" occurs when the virtual camera is far from the reference camera positions. In between these cases, arbitrary access patterns usually appear as lines like the one denoted as "C". Similar findings for a hierarchical prediction structure in two dimensions have been made in [TG03].

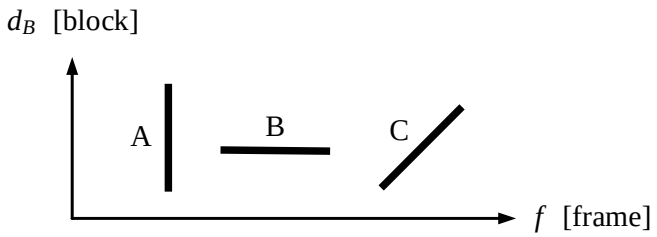

Figure 4.9: *Typical access patterns for rendering a novel view from an image-based data set. See the text for an explanation.*

Random access to a captured frame (case A)

To model the mean decoding complexity of a whole frame access, Equations (4.21)-(4.24) have to be modified to consider neighboring blocks within the same frame. This modification covers $a_{m,n} = a_{m,n-1}$ if $n > m$ and $m \neq 0$ to consider the fact that neighboring blocks are decoded and $a_{m,n} = 1$ if $m = 0$ as all blocks in the requested row are to be decoded:

$$a_{m,n} = \begin{cases} a_{m,n-1} & \text{if } n > m; m \neq 0 \\ 1 & \text{if } m = 0 \\ a_{m-1,n} \cdot (1-\alpha) & \text{if } m \neq 0; n = 0 \\ j_{m,n} & \text{else.} \end{cases} \tag{4.25}$$

Without considering the SKIP modes, the decoding complexity for a random frame access can be written as:

$$C'_{FA} = \frac{1}{N \cdot U} \cdot \sum_{f=0}^{N-1} \sum_{l=1}^{f} \sum_{t=0}^{U-1} a_{l,t}. \tag{4.26}$$

Here $U = N_x / B_x$ is the number of blocks within one image row. The complexity caused by the INTRA only encoded frame is:

$$C_{GFA} = \frac{1}{N \cdot U} \cdot \sum_{f=1}^{N-1} \sum_{t=0}^{U-1} a_{f,t}. \tag{4.27}$$

Again, similar to (4.19) the resulting complexity C_{FA} for a random frame access can now be written as:

$$C_{FA}(N, \alpha, \beta, b) = C_{GFA} + \left(C'_{FA} - C_{GFA}\right) \cdot \left(1 - \beta \cdot (1-\alpha)\right). \tag{4.28}$$

Figure 4.10 shows the comparison between a theoretical evaluation using (4.28) and experimental results for a GOP size of N=10 frames and different SKIP rates β. Note that the complexity is measured in decoded pixel per rendered pixel, i.e., all pixels in all requested blocks are assumed to be needed for view interpolation.

Random access using case B

For case "B" in Figure 4.9 more pixels have to be decoded than are actually requested. The modification of the decoding probabilities (4.21) to calculate the mean complexity C'_{FB} is as

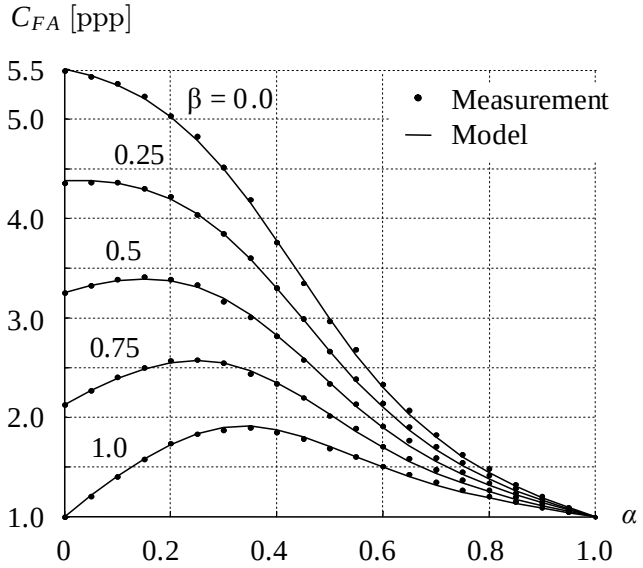

Figure 4.10: *The decoding complexity C_{FA} as a function of α and β for a GOP size of N=10 frames from (4.28).*

follows and reflects the fact that all requested blocks have a decoding probability of 1 (in case $n = 0$):

$$a_{m,n} = \begin{cases} 0 & \text{if } n > m \\ 1 & \text{if } n = 0 \\ j_{m,n} & \text{else.} \end{cases} \tag{4.29}$$

Similar to (4.22)-(4.24) the resulting complexity C_{FB} for a random access for case "B" is derived as (using (4.29) instead of (4.21)):

$$C'_{FB} = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{l=0}^{f} \sum_{t=0}^{l} a_{l,t}; \tag{4.30}$$

$$C_{GFB} = \frac{1}{N} \cdot \sum_{f=0}^{N-1} \sum_{t=0}^{f} a_{f,t}; \tag{4.31}$$

$$C_{FB}(\alpha, \beta, N, b) = C_{GFB} + \left(C'_{FB} - C_{GFB}\right) \cdot \left(1 - \beta \cdot (1 - \alpha)\right). \tag{4.32}$$

Figure 4.11 shows the comparison between a theoretical evaluation using (4.32) and experimental results for a GOP size of N=10 frames and different SKIP ratios β. Note that the complexity for low α and large β can fall below 1. This is less than for INTRA only encoding because blocks to be decoded share the same references in neighboring frames. Again, it is assumed that all pixels in a requested block are used for rendering.

Random access for arbitrary virtual views

In image-based rendering systems access patterns like the one in case "B" in Figure 4.9 are dominant when the virtual camera is far from the actual positions of the reference cameras.

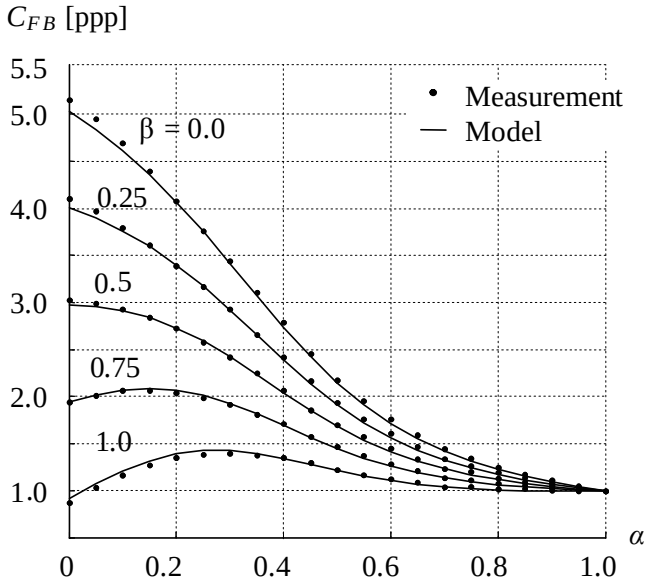

Figure 4.11: *The decoding complexity C_{FB} as a function of α and β for a GOP size of N=10 frames from (4.32).*

However, access pattern "C" can be interpolated using (4.28) and (4.32) according to the slope of the line in the f-d_B plane. For simplicity, in the following experiments (4.32) is used to approximate the decoding complexity while Equations (4.10)-(4.13) are evaluated to determine the decoding complexity for virtual views in real experiments. To only consider the number of actually needed pixels for rendering, a correction has to be introduced:

$$C_{FC}\left(N, \alpha, \beta, b, B_x, B_y\right) = \frac{C_{FB}}{\gamma}. \tag{4.33}$$

Here, γ is the pixel render versus request ratio which is the mean number of pixels used for view interpolation divided by the number of pixels that are actually requested (not considering dependencies) for a single access:

$$\gamma = \frac{\texttt{[number of pixels used for rendering]}}{\texttt{[number of blocks requested]} \cdot B_x \cdot B_y}. \tag{4.34}$$

Figures 4.12 and 4.13 show the theoretical and practical results for the decoding complexity C_{FC} for arbitrary virtual views and different values of β and block sizes B_x. To evaluate the theoretical models for decoding arbitrary virtual views, a concentric mosaic rendering system is used to determine random access patterns (γ values as well as column and row coordinates of block requests) during online operation for a real system and recorded user trajectories. Also random views are generated and corresponding frame and block requests are recorded. The maximum amount of translation between subsequent views is set to be the size of one 8×8 pixel block projected onto a plane at the mean depth of the scene (resulting in a step size of approximately 5cm in the experiments). Note that the distribution of INTRA, INTER, and SKIP modes are chosen according to the following procedure rather than random (this differs from the procedure in the former sections):

1. Disparity compensation on the original images of a GOP is performed.
2. According to α and β, INTRA blocks and INTER/SKIP blocks are distributed over all blocks as follows:
   - INTRA-mode is assigned to the fraction α of blocks introducing the biggest MSE after disparity compensated prediction.
   - INTER-mode is assigned to the fraction β of the remaining blocks which introduce the biggest MSE after disparity compensated prediction.
   - All other blocks are encoded in SKIP-mode.

This procedure ensures almost RD optimal compression as will be shown in Section 5.3.1 on page 116 while α and β can be freely adjusted.

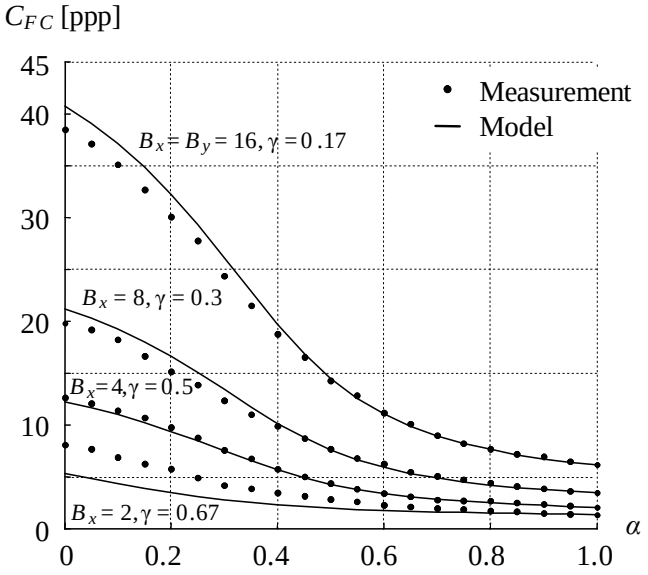

Figure 4.12: *The decoding complexity C_{FC} as a function of the INTRA ratio α and the block size B_x from (4.33). The GOP size is set to N=13, β=0.*

The models introduced in the former sections give an analysis of the decoding complexity of compressed image-based scene representations using hybrid video coding concepts. Additionally, a procedure for encoding group of pictures with predefined values for the INTRA ratio α and the SKIP ratio β is discussed. In the next sections a rate-distortion model is reviewed, the transmission rate model is introduced, and the final RDTC model will be evaluated.

4.3 The rate-distortion model

In this section a brief review of the rate-distortion model introduced in [Gir87] is given. The efficiency analysis of motion compensation presented in [Gir87] relates the power spectral density $\varphi_e(\omega_x,\omega_y)$ of the residual error to the accuracy of motion compensation captured by

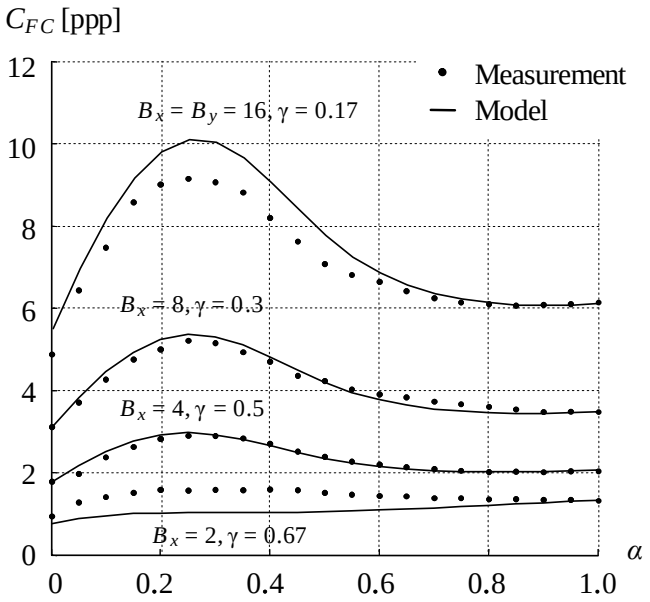

Figure 4.13: *The decoding complexity C_{FC} as a function of the INTRA ratio α and the block size B_x from (4.33). The GOP size is set to N=13, β=1.0.*

the probability density function of the disparity error [Gir00]:

$$\begin{aligned}\varphi_e(\omega_x,\omega_y) = \varphi_s(\omega_x,\omega_y)\cdot\left(1+\|F(\omega_x,\omega_y)\|-2\cdot\Re\left(F(\omega_x,\omega_y)\cdot P(\omega_x,\omega_y)\right)\right)\\ +\varphi_n(\omega_x,\omega_y)\cdot\|F(\omega_x,\omega_y)\|^2\end{aligned} \tag{4.35}$$

where (ω_x, ω_y) are the horizontal and vertical frequencies. φ_s and φ_n are the power spectral densities of the input signal and the residual error, respectively. F is the frequency response of the loop filter while P denotes the 2D Fourier transform of the disparity error probability density function. $\Re$ stands for the real part of a complex number. In [Gir87] three cases for F are distinguished:

1. $F(\omega_x,\omega_y)=0$ INTRA encoding (no prediction signal)
2. $F(\omega_x,\omega_y)=1$ using no spatial loop filter (suboptimal)
3. $F(\omega_x,\omega_y)=P^*$ optimum loop filter (approximated)

For simplicity, it is assumed that no loop filter is used (cases 1 and 2) in the following. This reduces the overall decoding complexity while sacrificing some coding performance, especially for full-pel motion compensation. For independent encoding of parts of the input signal (case 1), the error and signal are set equal: $\varphi_e=\varphi_s$ as $F(\omega_x,\omega_y)=0$. For case 2, with the substitution Λ=(ω_x,ω_y) the residual error power spectral density for motion compensated prediction can be written as:

$$\varphi_e(\Lambda) = 2\cdot\varphi_s(\Lambda)\cdot\left(1-\Re\left(P(\Lambda)\right)\right)+\theta \tag{4.36}$$

where θ represents spatially invariant noise with constant power spectral density $\varphi_n(\omega_x,\omega_y)$=$\theta$ and can take on any positive value. The rate and distortion for independent coding of the

signal (INTRA encoding) can be written in parametric form and independent from P:

$$D(\theta) = \frac{1}{4\pi^2} \cdot \iint_{\Lambda} \min\left[\theta, \varphi_s(\Lambda)\right] d\Lambda \tag{4.37}$$

and

$$R_{INTRA}(\theta) = \frac{1}{8\pi^2} \cdot \iint_{\Lambda:\varphi_s(\Lambda)>\theta} log_2\left[\frac{\varphi_s(\Lambda)}{\theta}\right] d\Lambda. \tag{4.38}$$

With (4.36) the rate for disparity compensated prediction (INTER encoding) can be written as

$$R_{INTER}(\theta, s) = \frac{1}{8\pi^2} \cdot \iint_{\substack{\Lambda:\varphi_s(\Lambda)>\theta \\ \text{and } \varphi_e(\Lambda)>\theta}} log_2\left[\frac{\varphi_e(\Lambda)}{\theta}\right] d\Lambda. \tag{4.39}$$

The underlying frame signal model of an image source is represented by an isotropic autocorrelation function:

$$A(\Delta x, \Delta y) = e^{-\omega_0\sqrt{\Delta x^2+\Delta y^2}} \tag{4.40}$$

where ω_0 is a parameter that represents the correlation between adjacent pixels. For numerical results in the remainder of this chapter, ω_0 is set to correspond to an average correlation factor of 0.93 (i.e., ω_0=-ln(0.93)). In typical video signals this correlation can be measured between horizontally and vertically adjacent pixels [Gir87]. The power spectral density of this signal is the Fourier transform of its autocorrelation function:

$$\varphi_s(\omega_x, \omega_y) = \frac{2\pi}{\omega_0^2} \cdot \left(1 + \frac{\omega_x^2 + \omega_y^2}{\omega_0^2}\right)^{-\frac{3}{2}}. \tag{4.41}$$

Sampling this signal on a lattice after band-limiting the frequencies $|\omega_x| \leq \pi$ and $|\omega_y| \leq \pi$ yields the space-discrete signal. Now, the missing part for evaluating (4.37)-(4.39) is the probability density function of the disparity error. This error reflects the accuracy of motion compensation which is limited due to the fact that motion vectors have to be stored or transmitted as side information with limited bitrate. It is assumed that the disparity error is only caused by rounding errors and is uniformly distributed over [-2^{-s},2^{-s}]x[-2^{-s},2^{-s}], (where s=1 for full-pel accuracy, s=2 for half-pel accuracy, s=3 for quarter-pel accuracy, and so on). Given s, the disparity error variance is

$$\sigma_\Delta^2 = \frac{2^{-2(s-1)}}{12}. \tag{4.42}$$

It has been shown [Gir00] that the actual distribution of the disparity error vector has not much influence on the RD performance as long as the variance $\sigma_\Delta^2$is the same. To simplify the analysis, the model now assumes that the vector valued disparity error Δ is isotropic Gaussian with variance σ_Δ^2. Then, the probability density function of the 2D disparity error is:

$$p_d(\Delta) = \frac{1}{2\pi \cdot \sigma_\Delta^2} e^{-\frac{\Delta^T\Delta}{2\cdot\sigma_\Delta^2}}. \tag{4.43}$$

Finally, P can be calculated using the 2D Fourier transform of the motion error probability density function $P(\Lambda) = \mathcal{F}\{p_d(\Delta)\}$ for a chosen motion vector accuracy s. As the system considered in this chapter only supports one dimensional disparity compensation, in the remainder of this work it is assumed that, though vertical disparity compensation is restricted, the disparity error probability density function is not affected.

The rate-distortion function for hybrid video coding incorporating the GOP size N and the INTRA-ratio α' can now be approximated using (4.37) to determine the distortion D and

$$R(\theta, N, \alpha, s) = \left(1 - \alpha'(N, \alpha)\right) \cdot R_{INTER}(\theta, s) + \alpha'(N, \alpha) \cdot R_{INTRA}(\theta). \tag{4.44}$$

The rate is assumed to be independent of the block size B_x. Furthermore, the rate is independent from the specific realization of the disparity vector field and therefore independent from b.

Note, that in practical coders encoding blocks in SKIP mode is very efficient in the rate-distortion sense mainly due to the possibility to skip significant coding overhead. For the Gaussian model in this analysis, when a constant distortion $D(\theta)$ for all possible encoding modes is maintained, R_{INTER} vanishes if $\varphi_e(\Lambda) \leq \theta$ and/or $\varphi_s(\Lambda) \leq \theta$ for the full range of Λ (which might hold for small regions in the input images but not in general). However, SKIP encoding is covered by the theory as a special case of INTER encoding. In the remainder of this chapter it is assumed that the rate does not depend on β.

4.4 A theory of RDTC optimal compression

In this section it is shown how the decoding complexity and transmission data rate can be incorporated into the RD model. Three cases which correspond to three different system setups are distinguished. The first setup is a remote navigation scenario implementing random access to arbitrary blocks of the compressed scene representation. The system does not use a cache, i.e., the client decodes a block, displays it or uses it for disparity compensated prediction and then reuses the memory for further decoding. The second scenario is similar except for the caching system. Here, the client keeps the compressed and transmitted bitstream in memory so that if a part of the bitstream can be reused, it has not to be transmitted again. In the third scenario caching in the pixel domain is performed, i.e., transmitted and decoded intensity values are stored at the client. Therefore, neither transmission nor decoding has to be performed twice per pixel block. Numerical results for the arbitrary random frame access pattern are also presented for this case.

4.4.1 The RDTC model without a cache (Case I)

The transmission data rate in a system that does not store already decompressed intensity values can be written for the single random block access using (4.17), (4.18), (4.38), and (4.39):

$$\begin{aligned} T_I(N, \alpha, \beta, b, \theta, s) = \; & R_{INTRA} \cdot \left((C_{SNCIII} - C_{GSNC}) \cdot \alpha + C_{GSNC}\right) \\ & + R_{INTER} \cdot \left((C_{SNCIII} - C_{GSNC}) \cdot (1 - \alpha)\right). \end{aligned} \tag{4.45}$$

Assuming small values for β (not covered globally by the RD model) and large GOP sizes (then $C_{GSNC} \ll C_{SNCIII}$ holds), the mean transmission data rate can be approximated by:

$$T_I(N, \alpha, \beta, b, \theta, s) \approx R \cdot C_{SNC}. \tag{4.46}$$

Figure 4.14 shows the RDTC plot (N=10, s=1, B_x=8, β=0, b is approximated using (4.8)) for the case with no caching when evaluating Equations (4.44), (4.37), (4.46), and (4.19). The rate R, the SNR, the decoding complexity C_I=C_{SNC} and transmission data rate T_I are shown (subscript I stands for case I). Without constraints on C_I and T_I the bitrate savings for constant PSNR is up to 20% compared to independent encoding (compare to the dotted lines in Figure 4.14 which represent constant PSNR). This gain reflects the efficiency of disparity compensated prediction as discussed in [Gir87]. In the case of common rate-distortion optimized compression the decoding complexity can be as high as 120ppp with a transmission data rate of over 100bpp for a high reconstruction quality.

4.4.2 The RDTC model with a bitstream cache (Case II)

When parts of the compressed scene representation are stored at the client side whenever data arrives, the mean decoding complexity is still the same as in Case I: C_{II}=C_I. But, as the bitstream for every block has to be transmitted only once, the transmission data rate is much lower. The transmission data rate for a single random block request in a scenario with bitstream caching is approximated according to (4.46) by replacing C_{SNC} with C_{SC}:

$$T_{II}(\alpha, \beta, N, b, \theta, s) \approx R \cdot C_{SC}. \tag{4.47}$$

Figure 4.14 shows an RDTC-plot for Case I (N=10, s=1, B_x=8, β=0, b according to (4.8)) and, additionally, there are three (dashed) lines of constant transmission data rate with bitstream caching plotted (Case II) using Equations (4.44), (4.37), (4.47), and (4.19) to determine RDTC measures.

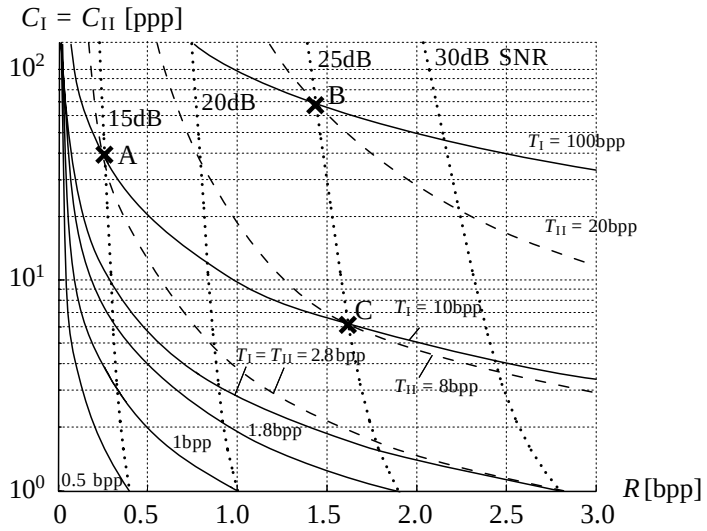

Figure 4.14: *RDTC-plot for a single block request for Case I and II (N=10, s=1, B_x=8, β=0, b according to (4.8)). The solid lines denote constant transmission data rate for Case I while the dotted lines represent constant quality. The dashed lines denote the lines of constant transmission data rate for Case II, for comparison. The marked points A, B, C are explained in the text.*

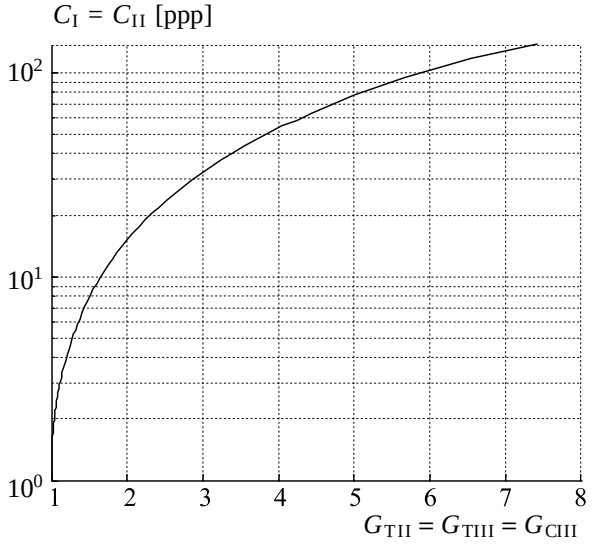

Figure 4.15: *The decoding complexity for Case I and Case II (compare to Figure 4.14) as a function of the caching gains G_{TII}=G_{TIII}=G_{CIII}.*

While bitstream caching has no influence on the decoding complexity, the gain (relative to a system without a cache) in terms of a reduced transmission data rate while the (storage) rate, distortion and decoding complexity remain fixed, can be expressed as G_{TII}=T_I/T_{II}. Figure 4.15 shows this gain with respect to the decoding complexity in Case I (no cache). For the marked position "A" in Figure 4.14, at C_I=C_{II}=40ppp, R=0.25bpp, SNR=15dB the transmission data rate is reduced by approximately 70% (from 10bpp to 2.8bpp, G_{TII}≈3.5). For the marked position "B" at C_I=C_{II}=70ppp, R=1.45bpp, SNR=25dB the transmission data rate is reduced by approximately 80% (from 100bpp to 20bpp, G_{TII}=5). The third marked position "C" at C_I=C_{II}=6ppp, R=1.65bpp, SNR=25dB yields a reduction of 20% for the transmission data rate (from 10bpp to 8bpp, G_{TII}=1.25). Naturally, the gain vanishes towards a decoding complexity of C_I=C_{II}=1ppp.

4.4.3 The RDTC model with a pixel domain cache (Case III)

In a system implementing a pixel domain cache, the decoding complexity for a single random block access pattern is C_{SC} from (4.24). The transmission data rate is equal to that in the case with bitstream caching: T_{III}=T_{II}. Figure 4.16 shows RDTC plots for the case of pixel domain caching (N=10, s=1, B_x=8, β=0) when evaluating Equations (4.44), (4.37), (4.47), and (4.24). The caching gains for both the transmission data rate G_{TIII} and the decoding complexity G_{CIII} compared to a system without cache are G_{CIII}=G_{TIII}=G_{TII} as shown in Figure 4.15.

Figure 4.17 (left) shows numerical evaluations of the theoretical RDTC model for a scenario with limited resources using constraints on the mean decoding complexity and transmission data rate (the dashed line in Figure 4.16 represents the RD curve with T_{III}≤T_{max}=10bpp and C_{III}≤C_{max}=10ppp). For comparison, Figure 4.17 (right) shows an operational RDTC plot from a real experiment using the same system parameters. At high rates a similar behavior as

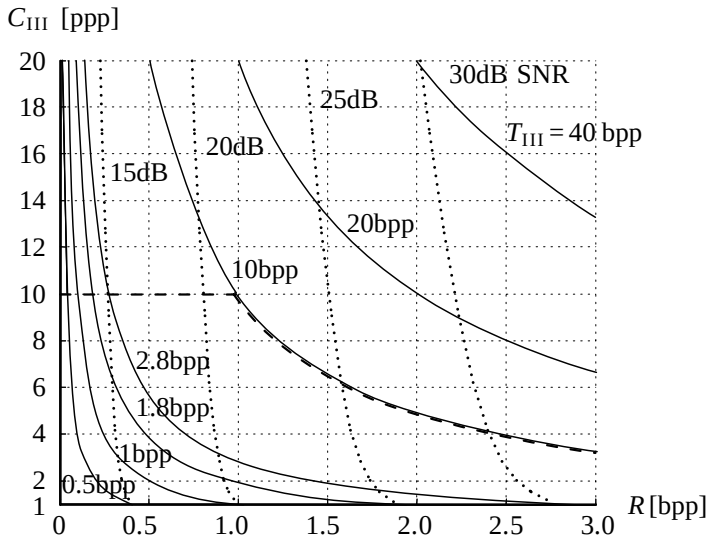

Figure 4.16: *RDTC-plot for Case III (N=10, s=1, B_x=8, β=0) and a random single block request. The solid lines are lines of constant mean transmission data rate while the dotted lines denote lines with constant PSNR.*

in theory can be observed. The quality gap between different scenarios is comparable. Note that theoretical results are evaluated using SNR while practical results are evaluated using PSNR. Both measures can be matched by a (in this analysis unknown and signal dependent) vertical shift.

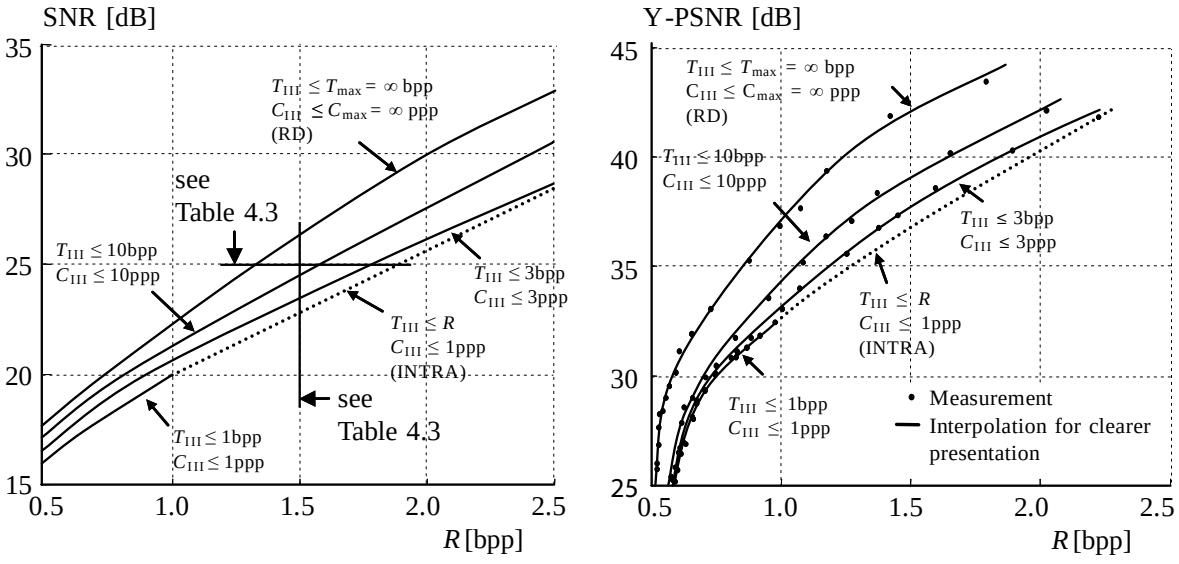

Figure 4.17: *RD curves for different decoding complexity and transmission data rate constraints using pixel domain caching (Case III, N=10, s=1, B_x=8, β=0, b according to (4.8)). Theoretical results (left) and measurements performed using a light field setup (right).*

Table 4.3 provides numerical results for RDTC optimization using various constraints for R=1.5bpp and SNR=25dB, respectively, as marked in Figure 4.17 (left). At R=1.5bpp for Scenario I (T_{III} is unconstrained while $C_{III} \leq C_{max}$=1ppp) an SNR of 23dB can be achieved while T_{III}=1.5bpp and C_{III} exactly matches its constraint. Note that this scenario corresponds to INTRA only encoding. In Scenario II no RDTC point can be reached. Here, the constraints are too strict so that no solution can be found for R=1.5bpp (although a solution at a lower rate R=1.0bpp can be found which yields an SNR of only 20dB). For Scenario III, a gain of 0.5dB can be observed compared to independent encoding at the same rate and at the cost of a higher transmission data rate and decoding complexity. For Scenario IV, an even higher gain can be achieved compared to independent encoding. RD optimized encoding in scenario V provides the highest SNR with significantly larger T_{III} and C_{III} compared to the other scenarios. Note that for scenarios III and IV INTRA encoding instead of RDTC optimized encoding would lead to a lower SNR as the available transmission data rate is not fully utilized. For the same scenarios, RD optimized compression would exceed the TC constraints resulting in a higher user perceived delay.

$R = 1.5$bpp

Scenario	T_{max} [bpp]	C_{max} [ppp]	R [bpp]	*SNR* [dB]	T_{III} [bpp]	C_{III} [ppp]
I (INTRA)	∞	1	1.5	23	1.5	1
II	1	1	-	-	-	-
III	3	3	1.5	23.5	3	2.5
IV	10	10	1.5	24.5	10	6.5
V (RD)	∞	∞	1.5	26.5	30	20

$SNR = 25$dB

Scenario	T_{max} [bpp]	C_{max} [ppp]	R [bpp]	*SNR* [dB]	T_{III} [bpp]	C_{III} [ppp]
I (INTRA)	∞	1	1.9	25	1.9	1
II	1	1	-	-	-	-
III	3	3	1.75	25	3	2.0
IV	10	10	1.6	25	10	6.0
V (RD)	∞	∞	1.3	25	25	20

Table 4.3: *Theoretical streaming performance of RDTC optimized scene representations.*

At a constant SNR of 25dB, INTRA encoding leads to the lowest T_{III} and C_{III} values while the storage rate R is large. Again, for Scenario II, no valid RDTC point can be found. RDTC optimization leads to the lowest possible storage rate while meeting the TC constraints for scenarios III and IV.

When considering the access pattern for arbitrary virtual view access from Section 4.2.3 on page 91, Figure 4.18 shows theoretical RD plots (N=13, s=1, B_x=8, β=0, γ=1, b is calculated using (4.7)) with TC constraints for the case with pixel domain caching. Equations (4.44), (4.37), (4.46), and (4.33) are evaluated. The rate R and PSNR for several constraints on the mean decoding complexity C_{III} and mean transmission data rate T_{III} are shown. From the

theoretical results in Figure 4.18 (left) it can be seen that for an SNR of 20dB, C_{max}=2ppp, and T_{max}=2bpp the rate R compared to independent encoding is reduced by 19% while the mean transmission data rate and therefore the user perceived delay is four times less than for RD optimized encoding (while increasing the rate by 25%). Comparable values can be obtained from the experimental results in Figure 4.18 (right) for a PSNR of, e.g., 35dB.

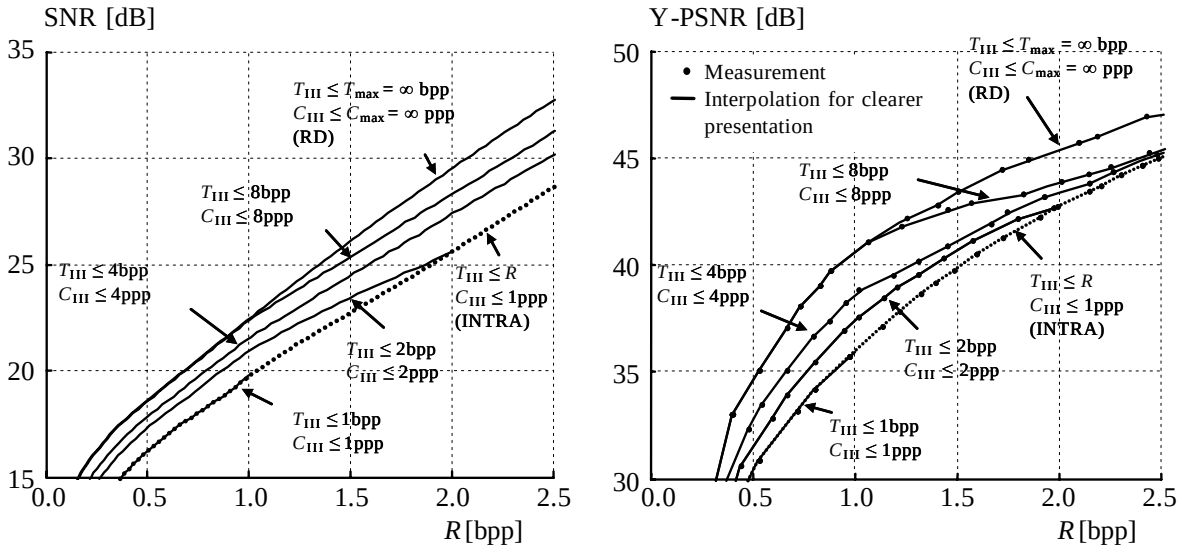

Figure 4.18: *RD curves for decoding complexity and transmission data rate constraints using pixel domain caching for whole virtual views (Case III, N=13, s=1, B_x=8, β=0, γ=1, b according to (4.8) (left) and (4.7) (right)). (left) Theoretical results and (right) measurements performed using a line light field setup.*

4.5 Issues with B pictures, 2D and hierarchical GOP structures

In this section, the use of bidirectional prediction and hierarchical GOP structures is discussed as well as the constraint to approximately rectified input sequences is relaxed. Without experimental validation, the basic principles for extending the theoretical findings are commented. A detailed analysis is left for future work.

Using B pictures In video coding so called B-pictures (bidirectional prediction) allow superior compression efficiency [Gir00]. B-pictures are most often predicted from the two nearest images (although these frames are most often not B-pictures themselves). By averaging the disparity compensated prediction signals, an even better prediction than for just one reference can be achieved. In [SNC05] a coding scheme based on bidirectional prediction for concentric mosaics is proposed. There, blocks in images between evenly spaced INTRA encoded images are predicted from two of the nearest INTRA images. A result of the work in [SNC05] is that the GOP size is limited to 6 neighboring images for the test sets they use (critically sampled concentric mosaics). This limitation is mainly due to the fact, that for view

centered scenes a significant amount of blocks in the B-pictures can not be properly predicted because their content is simply not visible in one of the neighboring INTRA encoded frames. In contrast, the scheme proposed in [ZL05] achieves a slightly better rate-distortion performance on the same test sets by using a single predictor from the nearest independently encoded image that is selected according to the rate distortion performance evaluated on a block basis during encoding. For object centered scenes and hierarchical GOP structures (as discussed in the next paragraph), the drawback regarding partially visible scene content is not that significant, thus B-pictures are successfully used (e.g., [MG00a, RKG07]).

Although used for the compression of image-based rendering data (but not for streaming in existing schemes), for random access, B-pictures introduce an additional overhead as more blocks have to be decoded per block request than for sequential coding because of the more complex dependency structure.

Branches in the prediction chain Hierarchical GOP structures involve branches in the prediction chain. I.e., one block can have dependent blocks in different frames, or vice versa, multiple blocks in different frames might reference the same block. To handle such cases, decoding complexity models introduced in this chapter can be extended. Consider the hierarchical prediction structure denoted in Figure 4.19. An independently encoded frame is chosen in the middle of a GOP (frame f=0). In the next level, two frames are encoded depending on the INTRA frame (frames f=2 and frame f=-2). In the next level 4 images depending solely on the frames in level 2 are encoded (frames 1,-1,3, and -3). For such a diadic

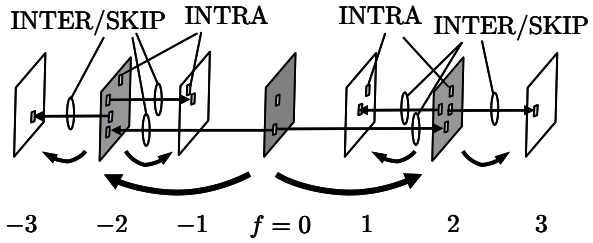

Figure 4.19: *An example of a three-level hierarchical GOP structure for line light fields. Bold arrows denote the prediction direction.*

hierarchical structure with L levels, the decoding complexity for β=0 in the case without a cache can be written as:

$$C_h\left(L,\alpha,b\right)=\frac{\sum\limits_{l=0}^{L-1}\left(2^l\cdot\sum\limits_{w=0}^{l}\left((2-b)\cdot(1-\alpha)\right)^w\right)}{\sum\limits_{t=0}^{L-1}2^t} \tag{4.48}$$

Here, the denominator is the actual number of images in the GOP. The nominator is the sum over the decoding complexities for a single block access in sequential GOPs of length $l\in[1,L]$ (as in, e.g., Equation (4.23)) weighted by the number of frames in that level. Generally, the decoding complexity is lower, especially for large GOP structures, compared to sequential structures with the same length. The impact of hierarchical structures on the rate-distortion trade-off is mainly characterized by the (physical) distance between dependent

frames. For hierarchical structures this distance is mostly larger than for sequential GOP structures thus providing a lower prediction accuracy. For a block access with a client cache or more complex access patterns, the model becomes far more complicated.

Prediction without rectification Up to now, only approximately rectified image sequences have been considered in the analysis. Disparity compensation was assumed to be done in horizontal (or only vertical) direction. Without this constraint, the decoding complexity for access on the block level as introduced in this chapter becomes very large even with small GOP sizes. Generally, without the restriction to rectified input images, a disparity vector points to an arbitrary position in a reference image. This might force the system to decode as much as 4 reference blocks (and their reference blocks) instead of only 2 for a single block request. However, the models introduced in this chapter can be extended to also cover the case for not rectified input images. E.g., the case of the single random block access without cache in Equation (4.17) with β=0 is extended to:

$$C_{nr}\left(N,\alpha,b_{41},b_{21}\right)=\frac{1}{N}\cdot\sum_{f=0}^{N-1}\sum_{l=0}^{f}\left(\left(4-3b_{41}-2b_{21}\right)\cdot\left(1-\alpha\right)\right)^{l} \tag{4.49}$$

where b_{41} is the single block ratio in the case of unrectified input images:

$$b_{41}=\frac{1}{B_e^2} \tag{4.50}$$

and b_{21} is the probability of having two reference blocks in a neighboring frame:

$$b_{21}=2\left(\frac{1}{B_e}-\frac{1}{B_e^2}\right) \tag{4.51}$$

where the probability of one component of the 2D disparity vector being a multiple of the block size B_x is approximated as $1/B_e$ assuming uniformly distributed disparity vectors. Note that the 2D disparity vector can be computed from a scalar disparity corresponding to the scene depth in the fully calibrated case. The statistical model in (4.23) can be extended in a similar manner by incorporating a two dimensional summation over the decoding probabilities in reference frames and adapting the probabilities themselves considering b_{41} and b_{21}. However, the calculation of the probabilities becomes very complex and a significant increase of the overall decoding complexity can be observed.

Group of pictures in 2D For more-dimensional capture geometries like light fields where the cameras are placed on a 2D grid, a hemisphere, or even freely in space, more efficient prediction structures have been proposed (e.g., [MG00a, ZL00]) than the one considered in this chapter. Such two dimensional hierarchical structures do not allow to restrict the input images to be rectified and also branches in the prediction chain have to be considered. Again, for the simple case of a single block request without a cache, Equations (4.49) and (4.48) can be combined. Assume, a 2D hierarchical structure as depicted in Figure 4.20 for a light field sampled on a regular grid. The numbers denote the hierarchy level the images belong to. Level 0 images are encoded independently. All higher level images are predicted from one of the nearest images in the next lower level (a similar scheme has been proposed in [ZL00]).

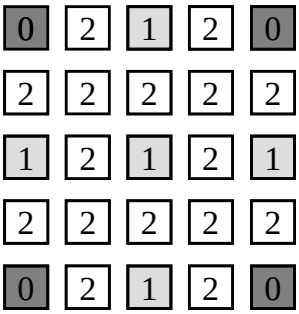

Figure 4.20: *An example of a 2D hierarchical GOP structure for 2D light fields sampled on a regular grid. Numbers denote the level l the image belongs to.*

Then, the number of frames in levels 0, 1, and 2 is 4, 5, and 16, respectively. The number of frames n_l within a level l is expressed by a function $n_l(l)$. With L=3 levels in this example, the double block reference ratio b_{21} and the single block ratio b_{41} for the case of unrectified input images, the decoding complexity can be written as:

$$C_{2Dh}\left(L,\alpha,b_{41},b_{21}\right)=\frac{\sum\limits_{l=0}^{L-1}\left(n_l(l)\cdot\sum\limits_{w=0}^{l}\left(\left(4-3b_{41}-2b_{21}\right)\cdot\left(1-\alpha\right)\right)^{w}\right)}{\sum\limits_{t=0}^{L-1}n_l(t)} \tag{4.52}$$

For a system with a cache and more complex access patterns, the analysis can be extended as indicated in the former two paragraphs.

4.6 Discussion

In this section some major issues regarding the limitations, the accuracy, insights, and applications of the models introduced in this chapter are discussed.

4.6.1 Limitations and accuracy evaluation

The major limitation of the models investigated in this chapter is that the input data set is assumed to be an approximately rectified image sequence. Though suitable for line light fields like concentric mosaics and for light fields that can be split into several line light fields, this prevents the system to exploit, e.g., hierarchical encoding structures. In Section 4.5 these issues have been addressed. Nevertheless, the models are still applicable at a loss of overall efficiency whereas the ability to control the RDTC measures remains.

Further, the RD models describe bounds on the rate-distortion trade-off that can be achieved while the complexity and therefore the transmission data rate models are mean values averaged over a large number of access events. The numerical results therefore only allow for qualitative conclusions.

In contrast to the assumptions made in this work, in real systems the block size has an

impact on the storage rate and on the decoding complexity (and therefore also on the transmission data rate).

A further simplification is that the reconstruction quality does not have an impact on the decoding complexity. In real systems the decoding complexity also depends on the number of entropy coded symbols, and therefore on the reconstruction quality. The additional error is about 20% in the relevant PSNR range from 30dB to 40dB according to experiments (also compare to [MG00a])).

The models only consider the very first block or view request of a streaming session. In real systems, when a user moves through a 3D environment, nearby virtual views also share reference blocks. Nevertheless, the scenario in the former analysis is regarded as the worst case. The gains predicted from the theory presented in this chapter are expected to be even higher in practical remote navigation scenarios (as discussed in the next chapter).

Further, the proposed theoretical models rely on the assumption that quantization does not have a significant influence on the realization of the disparity vector field. I.e., the disparity vector field found using the original reference frames should be exactly the same as when using the reconstructed frames. Intuitively, this assumption holds for high rates and whenever highly textured frames are captured.

While the models for a single block access pattern work rather accurately, for arbitrary virtual view access patterns, the error of the theoretical models for small block sizes is due to noisy disparity fields. For medium block sizes the maximum error is up to 10% for $N<15$ in according experiments. For large block sizes the model overestimates the real distribution due to border effects (the disparity is constrained for blocks near the frame boundary) when N is chosen too large.

A further source for model errors is the oversimplification of the impact of the realization of the disparity field to the signal dependent parameter b. The fact that b is estimated independently from the INTRA block distribution (the single block reference ratio is estimated including all INTRA blocks) contributes to the complexity estimation error. A further source for model inaccuracies is the way the INTRA and INTER/SKIP mode positions are chosen in the experiments. A random distribution would allow for a more accurate decoding complexity model while the error-based distribution as used in Section 4.2.3 on page 91 achieves a higher encoding efficiency.

4.6.2 Lessons learned from the theoretical analysis

From the theoretical analysis of the RDTC space derived in this chapter one can see that

- the adaptation to scenario specific parameters like available transmission rate and decoding capabilities gives a gain in terms of reduced storage rate or better quality compared to independent coding. Compared to rate-distortion optimal encoding a gain in terms of reduced user perceived delay can be observed while sacrificing some RD coding efficiency. The gains for transmission data rate and decoding complexity constrained optimization depend mainly on the encoding parameters summarized in Table 4.1.
- the caching strategy plays an important role for the overall system performance. Caching the received bitstream or even caching already uncompressed reference image parts at the receiver side gives gains up to a factor 5 for simple single block access

with previously empty cache. For more complex access patterns the gains are even larger (e.g., about factor 10 for arbitrary virtual view requests).

- pure RD optimization and independent encoding are border planes in the RDTC space.

4.6.3 Application

The introduced models can be used to analyze and design remote walkthrough applications using image-based scene representations. According to (4.2) constraints on T and C can be calculated as:

$$T \leq T_{max} = \frac{(t_d - t_{rtd}) \cdot r_{max}}{V} \text{ and } C \leq C_{max} = \frac{t_d - t_{rtd}}{t_C \cdot V}. \tag{4.53}$$

For instance, the encoding process can then be parameterized according to a desired RD trade-off, determined by (4.44), (4.37), (4.46), and (4.33) using the constraints in (4.53). Or, given a desired system response time t_d, then r_{max}, t_C, and the corresponding viewport size (V) can be traded off against the reconstruction quality for a given round trip delay t_{rtd}. Note that t_d specifies the response time of a remote walkthrough system using image-based scene representations. For $t_{rtd} = 0$ the mean frame rate can be determined by $1/t_d$.

4.7 Summary

In this chapter a theoretical framework for RDTC optimized compression and streaming of image-based scene representations based on hybrid video coding concepts is presented. The conventional rate-distortion optimization is extended to a trade-off between the rate R, the distortion D, and scenario specific constraints like the available transmission data rate T and the decoding capabilities of a client device (captured by the decoding complexity C). The RDTC system measures can be modeled from the encoding parameters used for offline encoding of groups of pictures and the signal properties (captured by the correlation between neighboring pixels and the displacement field properties captured by the single reference block ratio b). Main encoding parameters are the quantization parameter q, the block size B_x, the ratio of INTRA encoded blocks α, and the ratio of SKIP blocks among non-INTRA blocks β.

The theoretical results are validated by real experiments and show that an adaptation to scenario specific properties can give significant gains compared to rate-distortion optimal encoding or independent coding either in terms of a reduced user perceived delay or higher coding efficiency. Additionally, the impact of a client side caching system is evaluated and different access patterns are investigated.

The main contributions of this chapter are the detailed analysis of the decoding complexity and the mean transmission data rate for remote access to arbitrary parts of compressed image-based scene representations encoded using hybrid video coding concepts. Such an analysis has not been carried out and reported in the literature so far. Though the rate-distortion model gives theoretical coding bounds instead of accurate measures related to real data sets, the RDTC analysis gives insights in the decoding structure and transmission data rate which may be important for further research on streaming systems for image-based rendering. Parts of the analysis are used in the next chapter for encoding of real data sets and

for an investigation on the real-time behavior of RDTC optimized compression for streaming of image-based scene representations.

5 RDTC optimized compression - Practical coding

In this chapter the practical aspects of compression and interactive streaming for densely sampled structured image-based scene representations is investigated. According to the streaming system described in Chapter 4 (Section 4.1 on page 77), a framework for parameter estimation for RDTC optimal compression is given. Compression results and experiments using a real-time streaming testbed are performed and evaluated. A comparison to heuristic approaches is given, where applicable. Additionally, the impact of a finite size client cache on the overall streaming performance is measured.

In Section 5.1 the streaming system discussed in the former chapter is reviewed and practical issues are addressed. Trained models for the RDTC system measures are introduced in Section 5.2. Optimization of the streaming performance with respect to the initial delay is investigated in Section 5.3 while in Section 5.4 the objective is to optimize the system with respect to the mean user perceived delay. The joint optimization of the initial and mean delay is considered in Section 5.5. In Section 5.6 a real-time testbed is introduced and experimental results using this testbed are discussed. An extension to the optimization framework using decoding complexity constrained compression is presented in Section 5.7. Sections 5.8 and 5.9 discuss and summarize the findings of this chapter.

5.1 System overview of the practical RDTC coder

The practical system considered in this chapter is based on the system described in Section 4.1 on page 77 in the previous chapter. A hybrid coder working on group of pictures of size N is used. INTER and SKIP block modes are predicted by full pel accurate disparity compensation on 8x8 pixel blocks. Color conversion from the RGB to the YCbCr color space, transform coding (Discrete Cosine Transform (DCT) [CF77]), H.263 like quantization on visually weighted transform coefficients [ITU00], and entropy coding (Huffman coding [Huf52]) are applied. Additionally, the practical scheme has to provide pointers into the compressed representation in order to support random access to the partial bitstream representing a single pixel block. As has been pointed out in [SKC03] these pointers can be stored together with the compressed bitstream but sacrifice the compression ratio. At low rates up to 30% of the bitstream are covered by the pointers. In the system considered in this chapter, however, the pointers are implicitly compressed within the bitstream as special end-of-block symbols which can be parsed when a compressed representation is loaded in the beginning of a streaming session. The additional computational overhead is small. The system parameters that are used to parameterize the encoding process in the remainder of this chapter are (compare to Section 4.1.3 on page 80):

- The **quantization parameter** q (deadzone quantizer).
- The **INTRA ratio** α which is defined as the ratio of INTRA encoded blocks in a GOP (except for blocks in the first frame which are all encoded in INTRA mode).

- The **SKIP ratio** β which is the ratio of SKIP blocks among all blocks not encoded in INTRA mode.
- The **single reference block ratio** b which is signal dependent and is defined as the ratio of non-INTRA blocks that have one grid-aligned reference block in a neighboring frame.

RDTC optimized compression is performed with respect to the mean response time defined in Equation (4.2) in Section 4.1.3 on page 81. Online streaming is performed as discussed in Section 4.1.2 on page 80.

5.2 Trained RDTC models

In this section trained models for the RDTC measures are motivated and described. These models are based on the encoding parameters introduced in the previous section and allow for global optimization of the encoding process. It is assumed that the client implements a sufficiently large cache (in the comparative experiments this corresponds to about three to four times the number of pixels in the virtual view).

5.2.1 A heuristic approach for RDTC optimization

A straight forward approach for RDTC optimization is the Lagrangian RD optimization additionally considering constraints on the transmission data rate $T \leq T_{max}$ and the decoding complexity $C \leq C_{max}$. In common RD optimization approaches, the problem is to minimize the distortion D (e.g., measured as the mean squared error) given a rate constraint $R \leq R_{max}$ measured in bit per encoded reference pixel [bpp]. When considering T and C with additional constraints the problem can be written as:

$$\min_{R \leq R_{max}; T \leq T_{max}; C \leq C_{max}} D\left(R, T, C\right). \tag{5.1}$$

This can be solved using Lagrangian multipliers to weight the rate versus the distortion, the transmission data rate, and the decoding complexity:

$$\min_{R \leq R_{max}; T \leq T_{max}; C \leq C_{max}} J \;\text{ where }\; J = D + \lambda_R \cdot R + \lambda_T \cdot T + \lambda_C \cdot C. \tag{5.2}$$

Given the weights λ_R, λ_T, and λ_C, the solution of Equation (5.2) corresponds to a solution to Equation (5.1) when the TC constraints are met and $R = R_{max}$.

5.2.2 Trained models for RDTC system measures

Choosing the Lagrangian multipliers in (5.2) is not a trivial problem since their values are signal dependent. Using global models for the RDTC system measures for every GOP, based on the encoding parameters, allows us to find a global solution using numerical optimization rather than solving (5.2) on a block by block basis as it is done in common video compression schemes. When the GOP size N is assumed to be fixed (e.g., determined by the capture geometry), the models for the RDTC measures can be trained depended on the quantization parameter q, the INTRA ratio α, and the SKIP ratio β. The training of the models has to be

done for each GOP separately to adapt to signal properties (intra and inter-image correlation, occlusions,...). To get good interpolation results, six (α,β,q)-points are defined that are located at the border of the (α,β)-space with the quantization parameter chosen as q=1 which provides a high resolution of the distribution of the transform coefficients. This distribution is used for the approximation of the rate and distortion measures. The finally chosen sample points lie at (α,β,q) = (0,0,1), (0.4,0,1), (0,0.4,1), (0.6,1,1), (0,1,1), and (1,1,1). To obtain rate and distortion values as well as the distribution of the quantized coefficients at the sample locations the following procedure is used:

- The motion vector field for a GOP is determined using block matching on the original image data. b is determined using Equation (4.7).
- For each of the (α,β,q)-points the block mode distribution is found as follows:
  - INTRA encoding is chosen for the fraction α of blocks where the biggest MSE after motion compensated prediction is obtained.
  - INTER mode encoding is selected for the fraction β of the remaining blocks introducing the biggest MSE after motion compensated prediction.
  - Remaining blocks are encoded in SKIP mode.
- After transform coding and quantization of the INTRA coefficients and the residual error coefficients, Huffman coding is applied. For each of the resulting representations the distribution of the quantized DCT coefficients is stored and the rate as well as the average distortion (MSE) is measured.

Note that the motion vector field is determined on the original data and only once per GOP that is to be encoded. The main assumption here is that quantization does not have an impact on the realization of the motion vector field. Visually weighted quantization is performed by multiplying the quantization parameter q with the quantization tables proposed in H.263 [ITU00]. In the following the used models are described in detail.

Modeling the storage rate R

Common empiric rate-distortion models that consider the INTRA ratio α (e.g., for video over lossy channels in [SFLG00]) assume the SKIP ratio β chosen optimally in the (storage) rate-distortion sense. Since the rate model should be a function of β to incorporate the transmission data rate and decoding complexity in the remainder of this chapter, a new rate model has to be developed. The choice of the sample points at the edges of the α-β-space (see Figure 5.1) suggests a convex combination with respect to β of a convex and in this analysis exponential relationship between the rate and the INTRA ratio α:

$$R(\alpha,\beta) = R_{\alpha=0}(\beta) + (R_E(1,1) - R_{\alpha=0}(\beta)) \cdot \left(1 - (1-\alpha)^{\varepsilon_1 \cdot (1-\beta) + \varepsilon_2 \cdot \beta}\right) \tag{5.3}$$

with

$$R_{\alpha=0}(\beta) = R_E(0,1) + (R_E(0,0) - R_E(0,1)) \cdot (1 - \beta^{\varepsilon_3}) \tag{5.4}$$

The three model parameters ε_1, ε_2, and ε_3 are content specific and are trained from the six sample points. Note that (5.4) models R with respect to β for α=0. Subscript E indicates that the corresponding value is taken from the training samples. Equation (5.4) is used to serve as an initialization for the convex combination of the curves for R with respect to α with β=0 and β=1, respectively.

To extend (5.3) to be a function of the quantization parameter q, a modified version of the ρ-domain model introduced in [HM01] is used as an approximation. The rate R for different quantization parameters q can be calculated from the distribution of transform coefficients at the sample positions that are stored during the training of the models. The relationship between the ratio of zeros $\rho(q_0)$, produced by quantizing using q_0, and the rate $R(q_0)$ is used to train the parameter κ in the following equation:

$$R(q) = \kappa \cdot R(q_0) \cdot (1 - \rho(q)) + R_0. \tag{5.5}$$

Here, R_0 is the mean rate that is spent to signal the motion displacement and the block mode decisions at the training sample position under consideration and can be determined offline. The relationship between ρ and q is determined from the discrete probability distribution $f(y)$ of the transform coefficients y:

$$\rho(q) = \sum_{|y|<2q} f(y). \tag{5.6}$$

Originally, (5.5) was defined for rate control purposes with respect to one single frame or block. In the system described in this chapter, the storage rate of a whole GOP is of interest. Nevertheless, the accuracy is sufficient, even with error propagation during disparity compensation. Once κ is known, $R(\alpha,\beta,q)$ is determined by first extrapolating the coding samples using (5.5) and then applying (5.3) to interpolate in the α-β-space. Figure 5.1 shows an example plot of the rate R measured in bit per pixel as a function of α and β for a GOP size of N=13 frames in CIF resolution (352x288), a block size of B_x=B_y=8 pixel, and the quantization parameter set to q=1 and q=4 (q is used as in H.263), respectively. The values are obtained by sweeping over α and β and performing encoding as proposed in Section 5.2.2). For comparison, Figure 5.1 also shows results from experiments that are conducted using a densely sampled outdoor scene as described in Section A.4 on page 162. The accuracy of the model is acceptable with a maximum absolute error of 0.15bpp in all experiments.

Modeling D

Using a similar reasoning as for the rate in the previous section, an exponential model is chosen for the distortion for a constant quantization parameter q:

$$D(\alpha,\beta) = D_E(1,1) + (D_{\alpha=0}(\beta) - D_E(1,1)) \cdot e^{-\alpha \cdot (\mu_1 \cdot (1-\beta) + \mu_2 \cdot \beta)} \tag{5.7}$$

with

$$D_{\alpha=0}(\beta) = D_E(0,0) + (D_E(0,1) - D_E(0,0)) \cdot e^{\mu_3 \cdot (\beta-1)} \tag{5.8}$$

and μ_1, μ_2, and μ_3, which are, again, trained from the six sample points. Similar to the rate model, Equation (5.7) is extended to be a function of the quantization parameter q by using a modified version of the ρ-domain model introduced in [HM01]:

$$D(q) = D_0 \cdot e^{-\eta \cdot (1-\rho(q))} \tag{5.9}$$

where D_0 is the variance of the source signal. The relationship between ρ and q is determined from (5.6). Once η is known, $D(\alpha,\beta,q)$ is determined by extrapolating the coding samples using (5.9) and then applying (5.7). Figure 5.2 shows an example plot of the distortion D

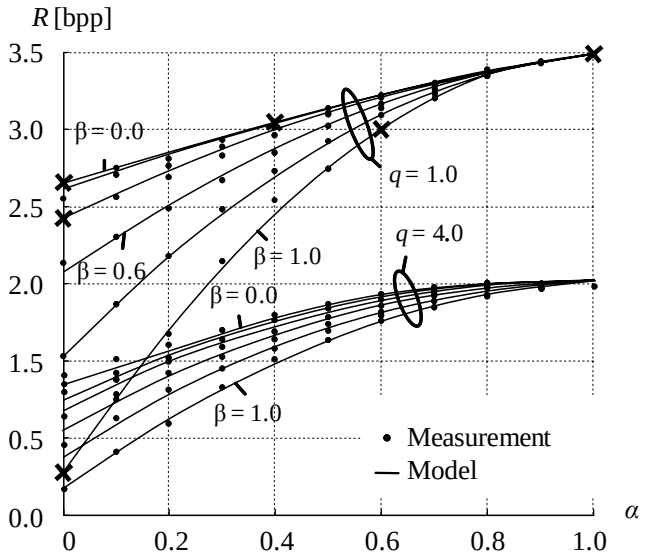

Figure 5.1: *The rate R measured in bit per pixel as a function of α and β for N=13, B_x=B_y=8, q=1 and q=4, respectively. The solid lines are calculated from the model in (5.3) and (5.5) and are compared to corresponding measurements (dots). The six crosses denote samples used for training of the model.*

measured in Y-PSNR (a PSNR measure only evaluated on the luminance component of the original and reconstructed image) as a function of α and β for a GOP size of N=13 (CIF), B_x=B_y=8 pixel, with q=1 and q=4, respectively. Again, the locations of the six sample points are shown as crosses. The worst case error of 1.9dB can be found in areas of the α-β-space that are not relevant as the RD trade-off becomes inefficient. The overall accuracy of the distortion model is sufficient for RDTC optimization purposes as will be seen later.

Modeling C

In the previous chapter a model for the decoding complexity C for whole virtual views as a function of α and β is presented. The single reference block ratio b, which is signal dependent, and the GOP size N are assumed to be fixed in that analysis. Using the probability $a_{m,n}$ for decoding a block with a position (m,n) relative to the requested block (0,0), the decoding complexity of a single random view access can be written as (compare to Equations (4.32) and (4.29) as derived in Section 4.2.3 on page 90)

$$C\left(\alpha,\beta,b,N,\gamma\right)=\frac{1}{N\cdot\gamma}\cdot\sum_{f=0}^{N-1}\sum_{t=0}^{f}\left(a_{f,t}+\left(\sum_{l=0}^{t}\left(a_{t,l}\right)-a_{f,t}\right)\cdot\left(1-\beta\cdot\left(1-\alpha\right)\right)\right). \tag{5.10}$$

Here, γ is the mean number of pixels needed for rendering divided by the number of pixels that are actually decoded per requested block for a single access as defined in Section 4.2.3 on page 91. γ is approximately a constant for a specific block size and rendering system and set to 0.3 for all subsequent experiments as determined by experiment. For the modeling

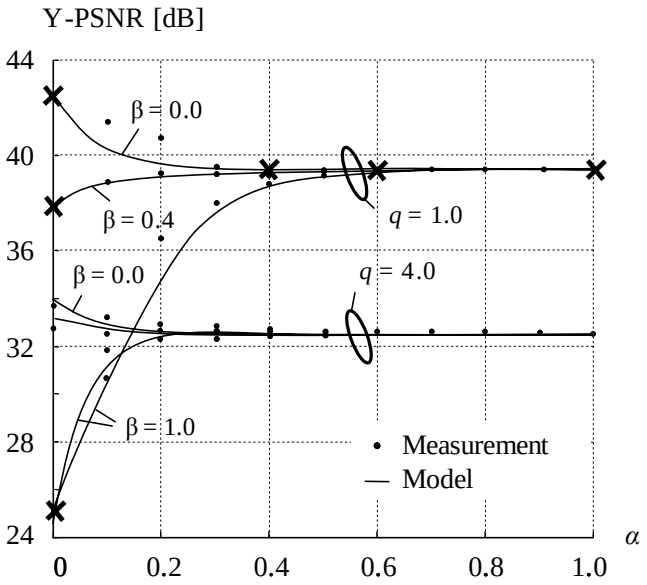

Figure 5.2: *The distortion (Y-PSNR) as a function of α and β for N=13 and B_x=B_y=8, with q=1 and q=4, respectively. The solid lines are calculated from the model in (5.7) and (5.9) and are compared to their corresponding measurements (dots). Crosses denote sample positions for model parameterization.*

process it is assumed that a sufficiently large cache is present, where sufficiently means that at least $N/2$ frames can be stored at the client.

Modeling T

The mean transmission data rate is approximately the weighted product of the decoding complexity C and the rate R (compare to Section 4.4 on page 96):

$$T(\alpha, \beta, q, b, N, \gamma) = \frac{R(\alpha, \beta, q) \cdot C(\alpha, \beta, b, N, \gamma)}{1 - (1 - \alpha) \cdot \beta}. \tag{5.11}$$

The denominator in (5.11) is important as the rate R is a mean over all blocks in the GOP whereas C only considers INTRA and INTER encoded blocks. Figure 5.3 shows the transmission data rate for an example GOP of size N=13.

5.3 Optimization with respect to the initial delay

In this section the models derived in the former sections are used to find the optimal encoder parameterization (α_{opt}, β_{opt}, q_{opt}) which leads to the minimum distortion for the given rate, decoding complexity, and transmission data rate constraints. In other words, given the available transmission bit rate T_{max}, the computational capabilities of a client machine C_{max}, and an overall storage rate R_{max}, the objective is to

$$\min\ D\ \text{ subject to }\ R \leq R_{max}, T \leq T_{max}\ \text{ and }\ C \leq C_{max}. \tag{5.12}$$

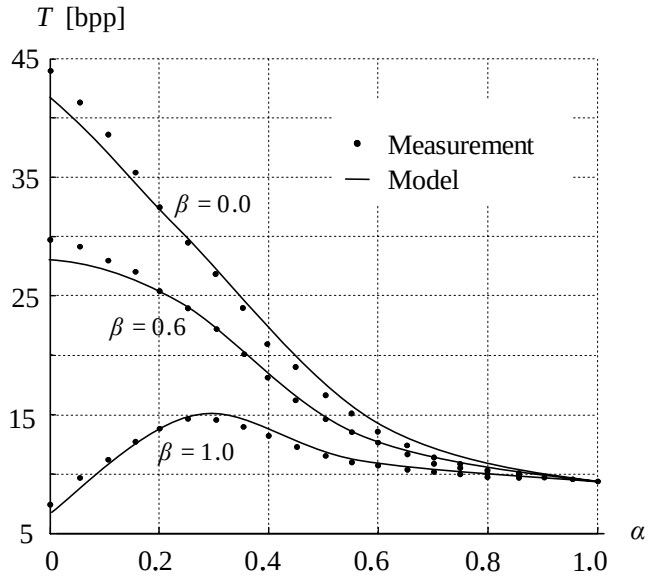

Figure 5.3: *The transmission data rate T measured in bit per requested pixel as a function of α and β. N=13, B_x=B_y=8, q=1, γ=0.3; The solid lines are calculated from (5.11) and are compared to their corresponding measurements (dots).*

C_{max} and T_{max} are calculated using (4.2) given the scenario and application specific parameters t_d, V, r_{max}, t_{rtd}, and t_c while the maximum storage rate R_{max} is arbitrarily chosen. The following procedure is used to perform RDTC optimization using the trained models (compare to Section 5.2.2 on page 110):

1. Block-based disparity estimation on the original images of a GOP is performed and a disparity vector field is produced. b is determined using Equation (4.7).
2. The GOP is encoded for the six sample parameter settings $\boldsymbol{z}_i$=[α_i β_i q_i]^T with i=1...6, producing model fitting values for R, D, T, and C. According to $\boldsymbol{z}_i$ INTRA and INTER/SKIP blocks are distributed as follows:
   - INTRA mode encoding is chosen for the fraction α of blocks where the algorithm calculates the biggest MSE after disparity compensated prediction.
   - INTER mode encoding is selected for the fraction β of the remaining blocks introducing the biggest MSE after disparity compensated prediction.
   - Remaining blocks are encoded in SKIP mode.
3. An optimal parameter set $\boldsymbol{z}_{opt}$=[α_{opt} β_{opt} q_{opt}]^T is found using numerical (constrained) optimization on the trained models according to the objective function (e.g., (5.12)).
4. According to the optimal values α_{opt} and β_{opt}, INTRA blocks and INTER/SKIP blocks are distributed as described in step 2 and entropy coding is applied to the quantized and weighted transform coefficients to form the optimized compressed representation.

The optimization problem in step 3 is numerically solved using Matlab. The objective functions and the constraint functions are continuous by design. While the rate function is monotonic, the distortion function is not, in general, monotonic with respect to the encoding parameters. The decoding complexity is independent from the quantization parameter q, but

not monotonic with respect to the INTRA ratio α. The same applies to the transmission data rate. Therefore, to ensure the convergence of the algorithm in the global minimum, the optimization procedure is performed four times with the initial parameter sets at the border of the α-β-space: (α,β,q)=(0,0,1), (0,1,1), (1,0,1), and (1,1,1). The optimal parameter set that gives the minimum value of the objective function is chosen. The execution time for the optimization is negligible.

5.3.1 Experimental results

The performance of the proposed model-based optimization is evaluated in Figure 5.4 and 5.5 by comparing it to a full search on regularly sampled RDTC points under limited computational power and available transmission rate.

Additionally, the approach introduced in the former sections is compared to a server-centric approach where the pixel blocks required to render a virtual view are decoded from an RD optimally encoded representation by the server. Then the rendered virtual view is compressed just-in-time using traditional RD optimization (using (5.2) with λ_C and λ_T set to zero) choosing the parameterization (λ_R) to meet the desired bitrate per virtual view (i.e., $R \leq T_{max}$). This allows for efficient exploitation of the redundancy between successive virtual views. From the client perspective conventional video streaming is performed while the server has a high computational load to carry.

The RD performance using conventional rate-distortion optimized hybrid video coding is also shown in Figure 5.4 and 5.5 to prove the validity of the proposed RDTC models and the block mode distribution.

Finally, the INTRA only encoded rate-distortion curve is given for comparison. For the experimental results in Figure 5.4, R and D are calculated as mean values while T and C are averaged over a large number of virtual view access simulations. Access patterns are recorded using an IBR rendering system based on concentric mosaic test sets as described in Appendix A.4 on page 162.

Figure 5.4 shows operational rate-distortion plots whereas Figure 5.5 (left) shows the transmission data rate T as a function of the storage rate R. T can be mapped to Y-PSNR in Figure 5.4 via the rate R. The same applies to Figure 5.5 (right) where the decoding complexity is measured in pixels that have to be decoded for every pixel that is requested (ppp). Note that all curves are valid for streaming a whole virtual view from the server to the client assuming the client's cache to be empty prior to the request. Table 5.1 gives exemplary RDTC values for the different constraints at a rate of R=1.0bpp as illustrated in Figure 5.4 and 5.5. For Scenario I where T and C are unlimited, the reconstruction quality of all representations except for INTRA only encoding are almost the same. The low reconstruction quality for the INTRA only case is due to the rate constraint. The transmission data rate as well as the decoding complexity is low for the INTRA only encoding and server centric approach (denoted as "Server c." in the tables and figures). As expected, RD optimized ("RD opt.") and RDTC optimized compression without constraints achieve almost identical results. The RDTC representation, which is encoded using the optimized encoding parameters determined using the trained models, performs very close to the numerical results predicted by the models ("t. models") and also is very close to the full search approach ("f. search") used to verify the model.

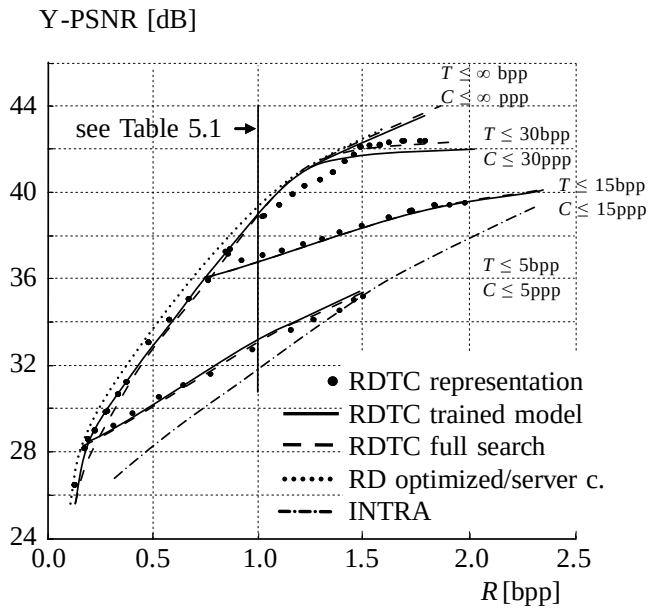

Figure 5.4: *Operational rate-distortion plots for different constraints on T and C using different optimization procedures. The solid lines denote the trained model as introduced in the former section. Dots represent the corresponding measurements using the optimal parameter set $\mathbf{z}_{opt}$ during encoding. Dashed lines are determined from a full search in the (α,β,q)-space. The dotted line gives the RD performance using Lagrangian optimization according to (5.1) and (5.2) by setting λ_C=0 and λ_T=0. Additionally, the dotted line denotes the RD performance of a server centric framework. The dashed-dotted line denotes INTRA only encoding.*

RDTC optimization starts to outperform conventional encoding when TC constraints are given. In Scenario II the transmission data rate and decoding complexity is limited to 15bpp and 15ppp, respectively. Now, again keeping the rate R constant at 1bpp, the RD optimized representation exceeds the transmission data rate constraint resulting in a significantly higher delay than specified and at the same time achieves a much higher PSNR than INTRA encoding which does not utilize the available transmission data rate and decoding complexity. RDTC optimization produces a bitstream that meets the requirements and achieves a 5dB better PSNR than INTRA only encoding with a loss of 2dB compared to RD optimal encoding.

For Scenario III the TC constraints are even more strict. Now, if the scene would have been compressed using conventional RD optimization, the user perceived delay would be over three times larger than required whereas the RDTC optimized approach meets the constraints. Compared to INTRA encoding a gain of 1dB can be observed in this scenario. Please note that the server centric approach always outperforms all other compression schemes at the cost of an intractable load of the server.

The 0.5dB loss in performance of the RDTC optimized representation compared to the RD optimized approach at rates up to 1bpp as shown in Figure 5.4 is due to the suboptimal distribution of INTRA, INTER, and SKIP modes as described in the beginning of Section 5.3 on page 114. This gap is considered to be small compared to the advantages like an

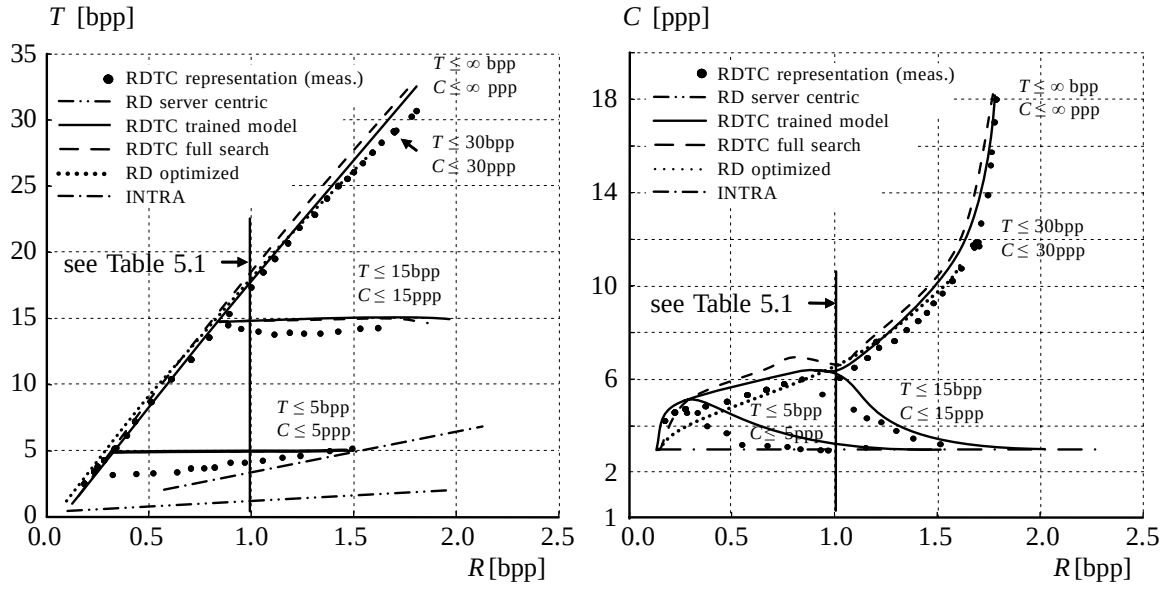

Figure 5.5: *Operational rate vs. transmission data rate and rate vs. decoding complexity plots for T and C constrained optimization. T and C can be mapped to distortion values in Figure 5.4 via the rate R.*

easy mapping from α and β values to block mode decisions. Please note that for clearer presentation, the scenarios are chosen so that the transmission data rate is the only limiting factor. In Figure 5.5 (right) the slope of the curves for $T\leq$15bpp and $C\leq$15ppp as well as for $T\leq$5bpp and $C\leq$5ppp decrease for R>0.25bpp and R>1bpp, respectively. This is due to the fact that the constraint for T avoids that C_{max} is actually reached.

		Scenario I $T_{max}=\infty$[bpp] $C_{max}=\infty$[ppp]			Scenario II $T_{max}=15$[bpp] $C_{max}=15$[ppp]			Scenario III $T_{max}=5$[bpp] $C_{max}=5$[ppp]		
	R [bpp]	Y-PSNR [dB]	T [bpp]	C [ppp]	Y-PSNR [dB]	T [bpp]	C [ppp]	Y-PSNR [dB]	T [bpp]	C [ppp]
RDTC	1.04	38.9	17.1	6.0	37.0	14.2	6.0	32.9	4.6	3.2
t. models	1.00	38.9	17.2	6.5	36.8	15.0	6.5	33.0	5.0	3.3
f. search	1.00	39.0	17.9	6.7	36.8	15.0	6.7	33.0	5.0	3.2
RD opt.	1.01				39.3	17.3	6.6			
Server c.	1.01				39.3	1.2	0.4			
INTRA	1.02				31.8	3.4	3.1			

Table 5.1: *Results for RDTC optimized compression with respect to the initial delay.*

5.4 Optimization with respect to the mean delay

In the previous section, the focus was on providing control over the initial delay between the request and display of the first virtual view. When the first view is completely available at the client, most of the subsequent views (hereafter called: second views) share reference data via the cache. In this section the performance of a streaming system is analyzed for the case that the user smoothly navigates through a scene and the cache starts to play an important role for the user experience.

For a virtual view that is selected near the position of the view that is currently displayed, most of the cached reference data can be reused. Figure 5.6 (right) shows the rendering geometry. View V1 is requested and processed as described in the previous sections. View V2 is selected near V1 and all the needed blocks to render this view lie next to the blocks already requested for view V1. The dominant access pattern is illustrated in Figure 5.6 (left). Blocks marked with a small black square are requested and used for rendering while white blocks have to be transmitted and decoded in turn. Hatched blocks are already in cache when the corresponding view is requested. In case (A) four blocks are requested and dependencies have to be resolved when the cache is empty. For the second nearby view either case (B) or case (C) is dominant depending on the actual position of V2. For case (B) and (C), in addition to the cached blocks, up to four further blocks have to be processed (in contrast to 10 blocks with an empty cache). This model is a rough approximation that includes rotation, forward, backward, and side motion of the virtual camera. It is derived from dominant access patterns that have been determined by extensive experiments on both compressed light fields and compressed concentric mosaics. It is based on the assumption that a user can move with a virtual step size that corresponds to the width of a pixel block when projected onto the image planes of the reference views on average. Similar findings have been reported in [TG03].

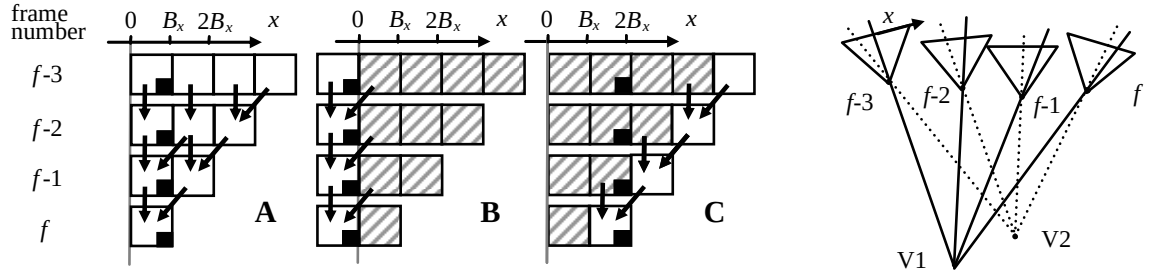

Figure 5.6: *(left) Example access patterns for a second view. First view (A) and further views near this first view (B), (C). (right) Top view for two nearby virtual views for case (A) (V1) and (B) (V2).*

5.4.1 Model parameterization for smooth navigation

While the models for the rate R and distortion D remain the same for the first and consecutive views, the models for C and T have to be adapted from the second virtual view on. Equation (5.10) is modified in order to consider those blocks that are not in the cache when a

nearby virtual view is requested. This is an approximation of the mean decoding complexity of the cases (B) and (C) in Figure 5.6 (left). The probability $a_{m,n}$ that a block at position (m,n) relative to the requested block has to be transmitted and decoded can be written as:

$$a_{m,n} = \begin{cases} 1 & \text{if } m,n = 0 \\ 0.5 & \text{if } m \neq 0; n = 0 \\ 0.5 \cdot a_{m-1,n-1} \cdot (1-\alpha) \cdot (1-b) & \text{if } m = n \\ 0 & \text{else.} \end{cases} . \tag{5.13}$$

These probabilities capture cases (B) and (C) in Figure 5.6 (left) simultaneously. I.e., in frame f the block marked with a black square has to be decoded in both cases (m, n=0), thus, the decoding probability is 1. In frames f-1 to f-3 the requested blocks ($m\neq$0, n=0) have not to be decoded in any case: For case (B) they have to be decoded, in case (C) not. As both cases are assumed to be equally probable, the decoding probability is set to 0.5. For ($m\neq$0,m=n) a similar reasoning can be done (compare to the theoretical analysis in the former chapter).

To calculate the decoding complexity C_S, Equation (5.10) is used with the probabilities from (5.13):

$$C_S(\alpha,\beta,b,N,\gamma) = \frac{1}{N\cdot\gamma} \cdot \sum_{f=0}^{N-1} \sum_{t=0}^{f} \left(a_{f,t} + \left(\sum_{l=0}^{t} a_{t,l} - a_{f,t} \right) \cdot (1 - \beta \cdot (1-\alpha)) \right). \tag{5.14}$$

Now, low INTRA ratios can lead to very low decoding complexity values compared to the case with an empty cache prior to the request. Figure 5.7 shows the decoding complexity C_S as a function of α and β. The transmission data rate T_S can be calculated according to Equation (5.11) by replacing C with C_S.

5.4.2 Experimental results

The performance of the proposed model-based optimization is compared to the server-centric framework described in Section 5.3.1 on page 116, to RD optimized encoding, as well as to independent encoding under limited computational power and available bitrate. The encoding procedure from Section 5.3 is used and the objective is formulated as:

$$\min\ D \ \text{ subject to } \ R \leq R_{max}, T_S \leq T_{S\max} \ \text{ and } \ C_S \leq C_{Smax}. \tag{5.15}$$

Here, T_{Smax} and C_{Smax} are constraints for T_S and C_S and are calculated similar to T_{max} and C_{max} for a desired mean delay for second views from Equation (4.2). T_S is shown in Figure 5.8 (left) as a function of the storage rate R. T_S can be mapped to Y-PSNR in Figure 5.4 via the rate R for corresponding encoding schemes (for the RDTC schemes, the unconstrained case in Figure 5.4 denotes the corresponding RD curve). The same applies to Figure 5.8 (right) where the decoding complexity is shown.

For the transmission data rate it can be observed that conventional RD optimized compression and INTRA encoding result in almost identical RT$_S$ curves. This suggests that pure RD optimized compression is the optimal choice as the distortion D is lower. The server centric framework still achieves the lowest transmission data rate. This is due to the fact that exactly the same number of pixels have to be transmitted and decoded as are displayed. For all other (non-server centric) approaches significantly more pixels have to be decoded

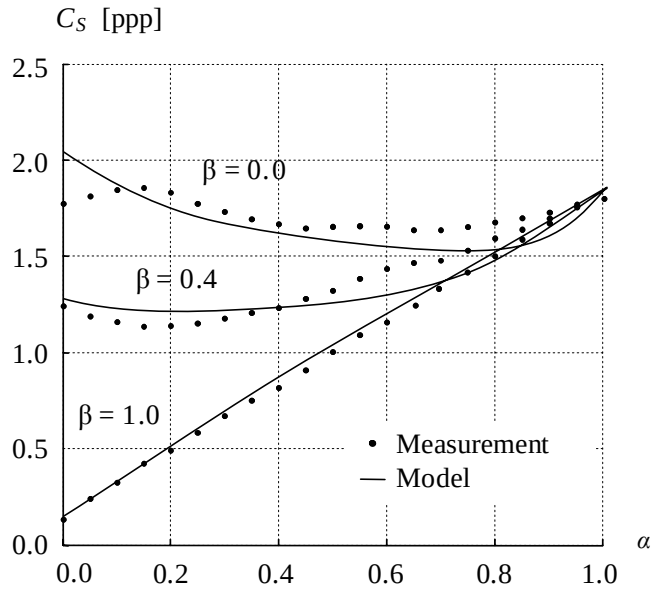

Figure 5.7: *The mean decoding complexity C_S for the case of an already decoded nearby virtual view in cache as a function of α and β; N=13, B_x=8, γ=0.3. The solid lines are calculated from the model in (5.14) and are compared to their corresponding measurements (dots).*

than are needed for rendering as γ=0.3<1.0). Note that the decoding complexity slightly decreases with increasing rate for the server centric approach. With a significantly higher quality of the INTRA blocks at higher rates, the probability of SKIP mode decisions slightly increases which decreases the decoding complexity for video decoding. But, as the storage rate increases faster, the transmission data rate also increases.

The decoding complexity shows that INTRA encoding performs worst while the server centric approach achieves very low complexity values. The arrows mark the maximum decoding complexity under the corresponding constraints for RDTC optimized compression. The fact that the constraint for C_S is never reached is, again, due to the choice of constraints in the examples. For clearer presentation, the constraints are chosen such that T_{Smax} is the limiting factor.

5.5 RDTC optimization for interactive streaming

In the previous sections, the RDTC models and encoding procedures for RDTC optimization of image-based scene representations have been introduced. Two main cases have been investigated. When the client's cache is empty prior to the request of the first virtual view, the constraints $T \leq T_{max}$, $C \leq C_{max}$, and $R \leq R_{max}$ are used to parameterize the streaming scenario. Trained models were introduced to determine the optimal encoding parameter triple $\boldsymbol{z}_{opt}$ which minimizes the distortion D while meeting the constraints, calculated according to Equation (4.2).

In the second case it was assumed that the cache is filled with reference data of a virtual view prior to the request for a nearby virtual view. Then, D has been minimized subject to

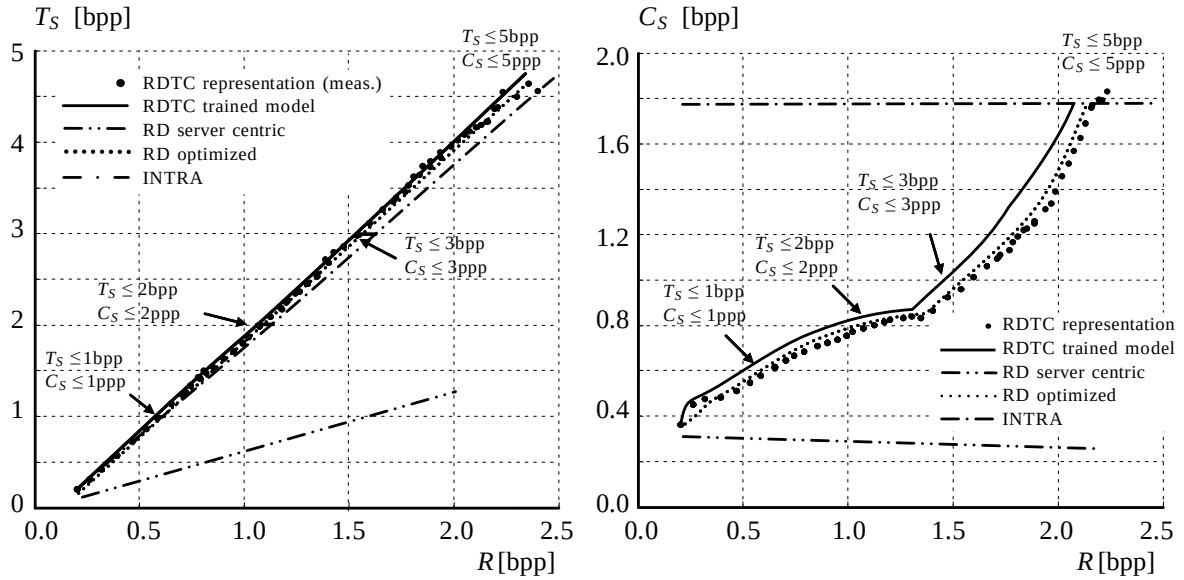

Figure 5.8: *Operational rate vs. transmission data rate (left) and rate vs. decoding complexity plots (right) for T_S and C_S constrained RD optimization in case of an already decoded nearby virtual view in cache.*

$T_S \leq T_{Smax}$, $C_S \leq C_{Smax}$, and $R \leq R_{max}$. In this section joint optimization with respect to both the initial and mean delay is discussed.

5.5.1 RDTC optimization for concentric mosaics and line light fields

For image-based scene representations, the independently encoded frame of a GOP can be shared across multiple GOPs as no playout order is dominant. Such a GOP structure for line light fields or concentric mosaics is shown in Figure 5.9 with frames 0 to 12 (first GOP) and -12 to 0 (second GOP). In this way, C and T can be calculated as for a N=13 frame GOP while the rate R is corrected compared to the model in Section 5.2.2. The fraction k of R that is allocated to the independently encoded frame is approximated by:

$$k = \frac{1}{N \cdot (1 - \beta \cdot (1 - \alpha'))} \quad \text{with} \quad \alpha' = \frac{\alpha \cdot (N-1) + 1}{N}. \tag{5.16}$$

Here, the term $1 - \beta \cdot (1 - \alpha')$ in the denominator is the fraction of blocks encoded in INTRA or INTER mode (no bits are allocated to the SKIP residual). α' is the INTRA ratio including the independently encoded frame. With I_G=2 GOPs sharing the INTRA frame for concentric mosaics or line light fields, the rate R_{CM} can be written as:

$$R_{CM}(I_G, \alpha, \beta, q) = (I_G - (I_G - 1) \cdot k) \cdot R(\alpha, \beta, q). \tag{5.17}$$

$R(\alpha,\beta,q)$ is the rate calculated from the RDTC models using frames 0 to 12 in Figure 5.9 assuming that the signal properties are equal for both of the combined GOPs. Especially for low α values and low bitrates, the gain of combining GOPs in (5.17) becomes significant.

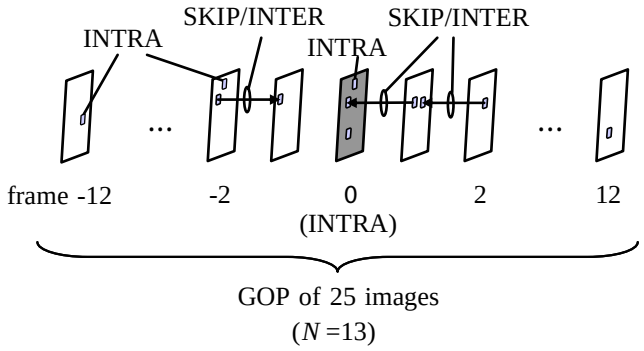

Figure 5.9: *GOP structure with example block modes for 25 images. Frame 0 is encoded in INTRA-mode. Arrows denote dependencies.*

5.5.2 Results for RDTC optimized compression of concentric mosaics

In this section the joint optimization with respect to the mean and initial delay for concentric mosaics is considered. In the exemplary scenario, one would like to provide a maximum distortion D_{max}, a maximum initial delay t_d, and at the same time would like to minimize the transmission data rate for second views T_S while C_S is unconstrained (assuming a powerful client machine). The optimization problem can be written as:

$$\min \ T_S \ \text{subject to} \ D \leq D_{max}, R_{CM} \leq R_{max}, T \leq T_{max}, \ \text{and} \ C \leq C_{max} \tag{5.18}$$

Figure 5.10 shows the results of an optimization using (5.18). For three different maximum distortion values D_{max}, the operational rate vs. decoding complexity plots are shown. Without loss of generality and for clearer presentation C_{max} and R_{max} are set to 20ppp and 3bpp, respectively, which means that both are unconstrained. Table 5.2 shows some numerical results taken from Figure 5.10. For 35dB Y-PSNR, and using T_{max}=15bpp, the transmission data rate for second views T_S becomes 1.8 bit per pixel (bpp). This point corresponds to a stream optimized using a rate-distortion trade-off solely (marked as "RD"). When T_{max} is decreased to 6.3bpp, then T_S increases to 3.1bpp. This configuration corresponds to independent encoding (marked as "INTRA"). For rates T_{max} between these two extreme values an R vs. T vs. T_S trade-off can be achieved. Please note the inverse course of T vs. T_S, and C vs. C_S with respect to the rate R. Table 5.2 also shows the RDTC values obtained with optimization using the trained RDTC-models. The corresponding encoding parameters are used to produce the final RDTC representation.

5.5.3 Comparison to encoding with multiple representations

Up to this point only a single compressed representation has been used for streaming. As proposed by [RG04b] for a light field streaming system, using multiple representations can improve the streaming performance. The main idea is that dependent on the cache situation at the client side, the transmission of INTRA or INTER/SKIP encoded blocks can be chosen online. To avoid ad hoc encoding at the server side, so called SI and SP frames [KK03] are used to make sure that different precoded streams have identical reconstruction to avoid error propagation during decoding (compare to Section 2.8 on page 26). Though this technique

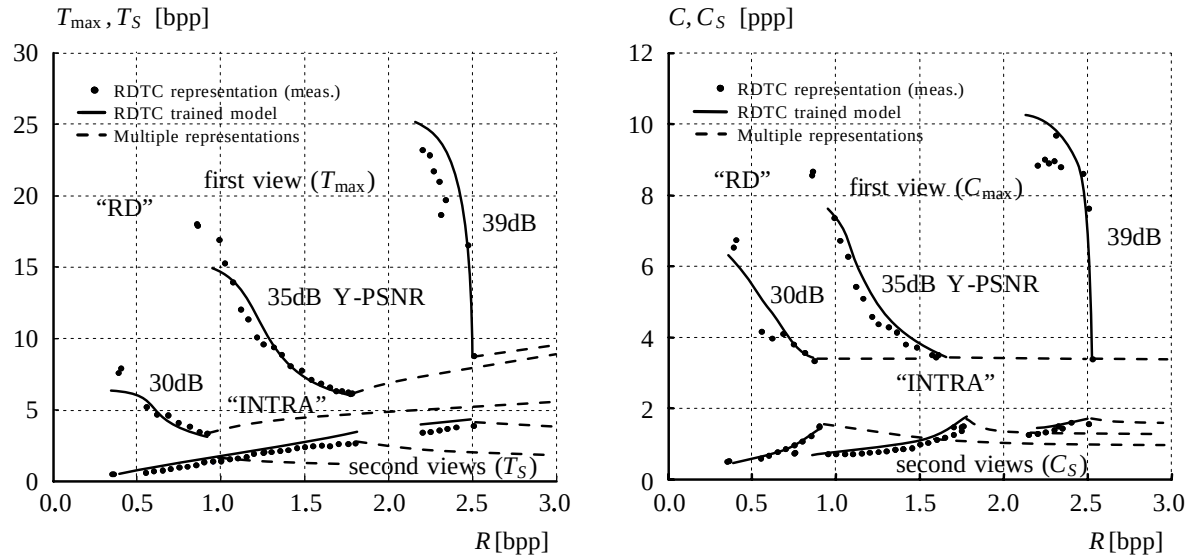

Figure 5.10: *Operational RC and RT plots for an RDTC optimization using (5.18). For different qualities the R vs. C_{max} and R vs. C_S as well as the R vs. T_{max} and R vs. T_S trade-offs are plotted. Measurements using the corresponding optimized configuration are shown as dots. The dashed lines denote curves obtained by using multiple representations as explained in the text.*

	RDTC representation	RDTC trained model	multiple representations	
T_{max}[bpp]	17.0	15.0	-	
T_S[bpp]	1.8	1.9	-	$R = 0.9$bpp
C[ppp]	8.0	7.5	-	(RD)
C_S[ppp]	0.8	0.8	-	
T_{max}[bpp]	6.3	6.3	6.3	
T_S[bpp]	3.1	3.8	3.1	$R = 1.8$bpp
C[ppp]	3.2	3.2	3.2	(INTRA)
C_S[ppp]	1.8	1.9	1.8	
T_{max}[bpp]	-	-	8.5	
T_S[bpp]	-	-	2.0	$R = 3.0$bpp
C[ppp]	-	-	3.2	
C_S[ppp]	-	-	1.5	

Table 5.2: *Performance of compressed representations at Y-PSNR=35dB for different rates R.*

provides flexibility with respect to the dependency structure at run time, three challenges arise:

- First, as multiple encoded representations have to be stored, the rate R is significantly larger than for other compression techniques.

- Due to the quantization of the prediction signal, the RD performance of SP blocks is worse than for conventional INTER blocks. Also the SI representation performs worse than conventional INTRA encoding in the RD sense. As a consequence, the RD performance of the RDTC optimized streaming system can not be reached.
- The second point is that online scheduling has to be performed to choose the ideal truncation of dependencies (i.e., the usage of the independently encoded stream).
- Third, the prefetching effect that is observed when using a single representation (either RD or RDTC optimized) is not exploited.

To compare to RDTC optimization, a system using three representations is implemented. The first and second representation is optimized as described in the former sections except that INTER and SKIP modes are replaced by their corresponding SP representations. A further difference is that a frame f is encoded dependent on frame f-1 in the first and dependent on frame f+1 in the second representation (two dependency directions). The third representation consists of independently decodable SI blocks wherever dependent encoding was performed in the first and second representation. For the first virtual view and pure rotation the SI representation is always chosen and transmitted when a requested block does not reside in the client's cache. For translational movement, to simplify the online scheduling process, the SP representations is transmitted dependent on the moving direction.

Figure 5.10 also shows the results for streaming using encoding with multiple representations with respect to (5.18) and the same settings as for RDTC optimization as described in the previous section. As expected, the decoding complexity remains constant for the first view (SI). With increasing bitrate also the decoding complexity of the "second views" decrease as more INTER/SKIP mode decisions can be utilized. With increasing rate the transmission data rate also increases as SI encoding is less efficient than conventional INTRA encoding. For second views, the transmission data rate decreases. The advantage of multiple representations encoding compared to RDTC optimized encoding is the constant worst case decoding complexity due to the third independently encoded representation and the better T_{max} vs. T_S trade-off at the cost of a very large rate.

Again, Table 5.2 gives some numerical results taken from Figure 5.10 for clearer presentation. The RDTC representation and encoding with multiple representations at R_{max}=1.8bpp achieve exactly the same results because only INTRA mode blocks are used for both techniques. When T_{max} is increased to 8.5bpp then the minimum T_S drops to 2.0bpp. Note that this value is approximately the same as for RDTC optimization at R_{max}=0.9bpp but with a significantly lower value for T_{max} (8.5bpp vs. 15.0bpp) and C (3.2ppp vs 8.0ppp) whereas C_S (1.5ppp vs. 0.8ppp) and R_{max} are bigger.

5.6 Real-time experiments

To evaluate the real-time performance of the proposed RDTC optimization framework, a streaming testbed is implemented. As illustrated in Figure 5.11 the testbed consists of an encoder, a server and a client, a channel simulator, a renderer, and a caching system. The encoder performs the RDTC optimized offline encoding on concentric mosaics and stores a compressed bitstream on the server. During online operation the user chooses a virtual view using the keyboard for translational movement and the mouse for rotational movement. The chosen view is requested and the server assembles the required bitstream. A pixel domain

cache (P-Cache) for already decoded pixel data is implemented. The server uses an index-table (P-Index) to keep track of the client's cache state. This prevents the server to transmit a compressed block if it is already available at the client. A least-recently-used displacement strategy is performed by the client which is mimicked by the server. The assembled bitstream is transmitted to the client via the channel simulator limiting the throughput. The client decodes the received data at the specified maximum decoding rate and updates the cache. The renderer performs nearest neighbor interpolation of light rays for view generation whenever the virtual camera rotates or translates, during still stand, bilinear interpolation is performed as described in [SH99] (compare to Section 6.2.1 on page 140). The testbed can generate random views, arbitrary motion trajectories, and can capture and replay motion performed by a human user. Such trajectories are then used to emulate user input to perform tests with varying system settings.

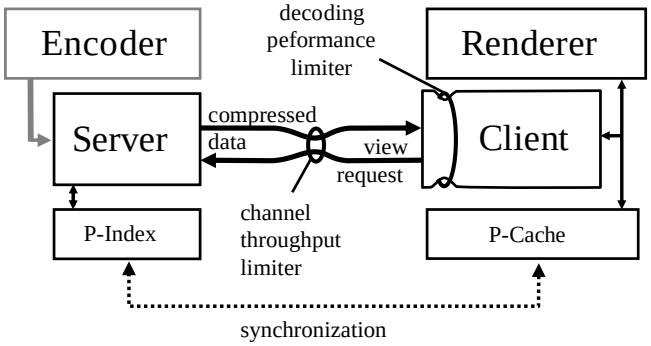

Figure 5.11: *The block diagram of the streaming testbed.*

The test data sets are described in Appendix A.4 on page 162. For the real-time experiments the indoor data set is used. Parameters are B_x=B_y=8, a maximum MSE of D_{max}=8.2 (40dB PSNR), a desired second view mean rate T_{Smax}=1bpp (resulting in 10fps at CIF resolution for a bitrate of 1Mbps), and a maximum second view mean decoding complexity C_{Smax}=1ppp. In the exemplary scenario, the minimization of the mean initial transmission data rate T with constraints on the distortion D, mean transmission data rate T_S, and mean decoding complexity C_S for second views can be written as:

$$\min \ T \ \text{ subject to } \ D \leq D_{max}, T_S \leq T_{Smax}, \ \text{ and } \ C_S \leq C_{Smax}. \tag{5.19}$$

5.6.1 Overall system performance

The performance of the considered streaming system is evaluated with respect to different modes of movement and the available pixel domain cache size. A target system capable of decoding 1 million pixels per second (1Mpps) and a maximum channel bitrate of 1Mbps is chosen. The virtual display is 320×480 pixels in size. Four different motion scenarios are defined: random views, rotation, translation, and trajectories that would appear in a first person 3D game. The latter is recorded during online operation of the testbed without any constraints from a user that repeatedly tries to locate a certain detail in the scene and also tries to hide behind obstacles (e.g., a window-frame or a tree in the test data sets). Several motion trajectories for each scenario are generated and replayed under different cache sizes.

A random view trajectory consists of 250 random positions and random viewing directions within the test set. The mean delay introduced by interactively transmitting these 250 views is measured with preemptied cache. Figure 5.12 shows the result.

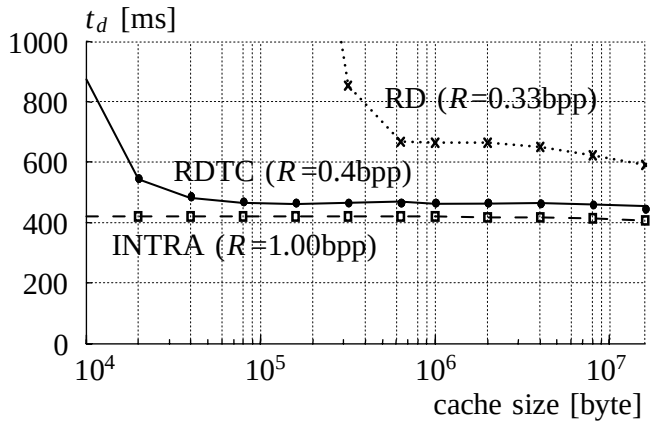

Figure 5.12: *Mean user-perceived delay for random views as a function of the pixel domain cache size at 40dB PSNR, a maximum bitrate of 1Mbps, and a maximum available decoding complexity of 1Mpps.*

Note that INTRA encoding performs best when the cache is switched off. Already with a small cache, RDTC optimized streaming leads to almost the same mean delay as INTRA encoding. RD optimized encoding has always the worst performance. Also note that the storage rate R is different for different optimization strategies as denoted in the figures. While the RDTC optimized stream has a slightly increased storage rate compared to RD optimized encoding, independent encoding leads to almost three times the file size than RD optimization. The same kind of experiment is performed for the remaining modes of motion and the results are shown in Figures 5.13-5.15. The target delay of 100ms for large cache

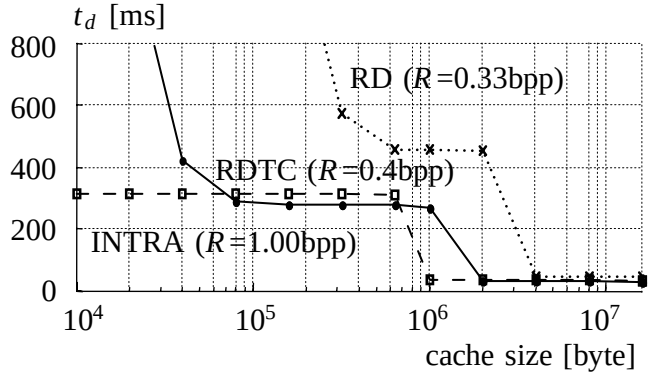

Figure 5.13: *Mean user-perceived delay for pure rotation as a function of the pixel domain cache size.*

sizes and translational movement can almost be achieved (see Figure 5.14). For the worst case delay illustrated in Figure 5.16, RDTC optimization for the target system achieves more

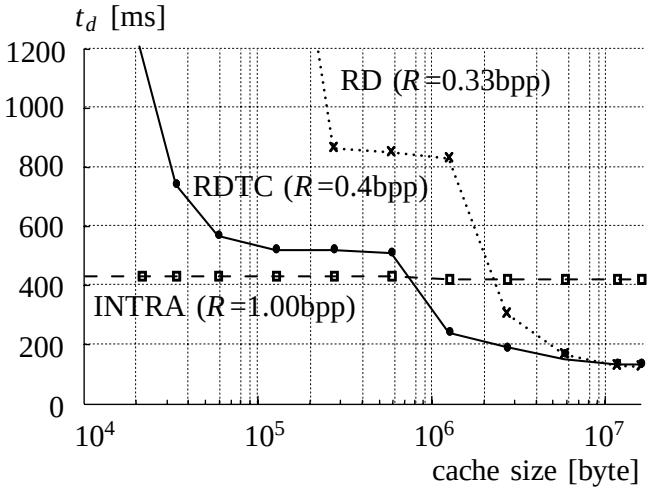

Figure 5.14: *Mean user-perceived delay for pure translation as a function of the pixel domain cache size.*

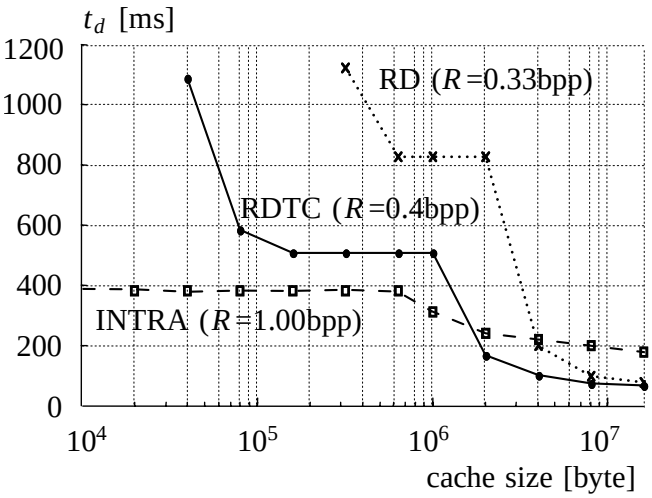

Figure 5.15: *Mean user-perceived delay for a trajectory consisting of translation and rotation as it can be found in first person 3D games.*

than 40% smaller values for the worst case delay than what is observed for RD optimized streams. The delay increases by 20% compared to INTRA encoding which, however, leads to a significantly larger storage requirement on the server.

5.6.2 Impact of the caching system

With a cache size of about 0.2 times the image raw data size (∼50kB) a significant gain in terms of reduced delay can be observed for RD and RDTC optimized compression. This is the cache size where the first blocks within the decoding process of a column can be reused.

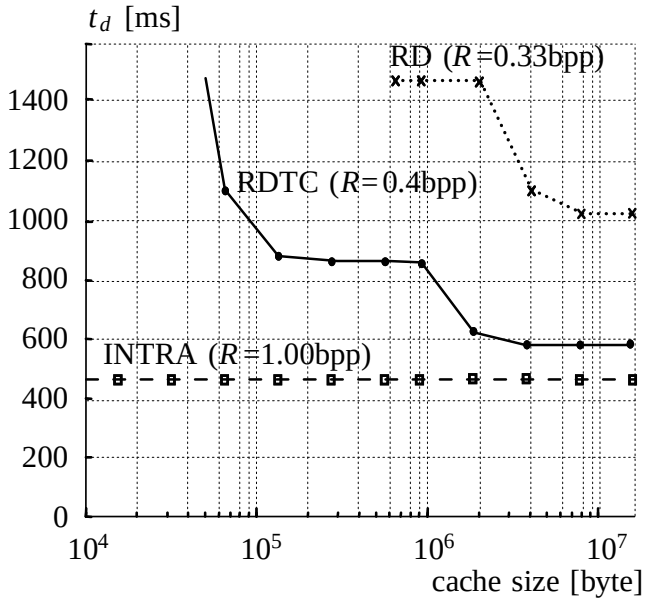

Figure 5.16: *Maximum user-perceived delay for a trajectory consisting of translation and rotation as it can be found in first person 3D games.*

There is another significant gain when the cache size is larger than 3-4 images ($\sim$1MB). This is when blocks between successive virtual views are shared via the cache. Note that approximately the number of blocks equivalent to 3 images have to be decoded to display one virtual view (corresponding to the pixel render/request ratio $\gamma \approx 0.3$ in (5.10) and (5.14)). For a pixel domain cache size of about 10MB or 30 images, there is no decrease of the user-perceived delay anymore.

5.7 Decoding complexity constrained disparity compensation

This section introduces a possible extension to the RDTC optimized compression framework discussed in the previous sections. Similar to rate constrained motion compensation [Gir94], complexity constrained disparity compensation can be performed. Formally, rate constrained motion compensation is the minimization of an overall distortion D (e.g., MSE) subject to a (disparity vector) rate constraint:

$$\min\ D\ \text{ with }\ R_{MC} \leq R_{max} \tag{5.20}$$

Where R_{MC} is the mean rate spent for encoding the motion vectors and R_{max} is the maximum allowed mean rate for encoding the motion vector field of a frame or GOP. Due to the fact that the decoding complexity of a compressed scene representation heavily depends on the actual disparity field realization and can be modeled by the single block reference rate b in the case of a sequential GOP structure as denoted in Figure 4.2, the formulation of rate constrained disparity compensation from Equation (5.20) can be extended to a joint rate-decoding complexity constrained disparity compensation. This can be thought of as trading

off rate and distortion for choosing a particular disparity vector for a block versus the single reference block ratio b. E.g., (5.20) can be rewritten as a Lagrangian optimization problem incorporating a decoding complexity constraint on a block by block basis:

$$\min\{j\} \quad \text{with} \quad j = d_{FC} + \lambda_r \cdot r_{MC} \quad \text{and} \quad C_{FC} \leq C_{max} \tag{5.21}$$

Here, d_{FC} is the distortion (after disparity compensated prediction) for choosing a particular disparity vector for the block under consideration, r_{MC} is the rate for coding this disparity vector. The global constraint $C_{FC} \leq C_{max}$ expresses that the mean decoding complexity C_{FC} is not allowed to exceed C_{max}. With this formulation and proper choice of λ_r and the quantization parameter q, the decoding complexity C_{FC} can be traded off against the rate and distortion. In the following experiment, a mean decoding complexity C_{FC} that is to be achieved, is chosen. With α=0, β=0, γ=1, and N=10, Equation (4.33) from Section 4.2.3 on page 91 is numerically solved for b. According to the obtained single block reference ratio b, the disparity vectors of those blocks are chosen to be multiples of B_x, which in turn introduce the lowest distortion.

Results for decoding complexity constrained disparity compensation are shown in Figure 5.17 and 5.18 for decoding an arbitrary virtual view from several GOPs of a concentric mosaic. The entropy coding (Huffman coding) is adjusted (trained) for every RDC point in the figures. Note that for increasing b the rate for disparity compensation decreases until $b<0.5$ while at the same time the quality almost does not degrade. The reason is that disparity vectors can be chosen as multiples of the block size without sacrificing the prediction accuracy too much, e.g., in areas with no texture but slight sensor noise. Figure 5.18 shows the distribution of disparity displacements for an example of unconstrained and constrained disparity compensation (increasing b). $p_{\Delta x}$ denotes the probability mass function of the disparity displacement. In the example, for b=0.05 (actually, b has some value $b>0$, because even for unconstrained disparity compensation, many disparity vectors are multiples of B_x). With C_{max} decreasing, the probability of disparity vectors that are multiples of B_x increases until for b=1.0 only disparity vectors that are multiples of B_x remain. With given values for α, β, γ, and N this disparity vector field then implements a minimum decoding complexity.

5.8 Discussion

In this section some major issues regarding the limitations, the accuracy, and insights of the models and results introduced in this chapter are discussed.

Limitations, accuracy and findings The major limitation of the coding system discussed in this chapter is, again, that the input data set is assumed to approximately be a rectified image sequence. In Section 4.5 this issue has been addressed. However, general scene representations that can be split into nearly rectified sub sequences can be encoded RDTC optimal with the introduced framework.

The accuracy of the practical models introduced in Section 5.2 is sufficient as shown in Section 5.3.1 by comparison to a full search on the parameter space (Figure 5.4). Also the block mode selection introduced in Section 5.3 is reasonable without sacrificing the coding performance too much (again, see Figure 5.4). The mode selection procedure allows for

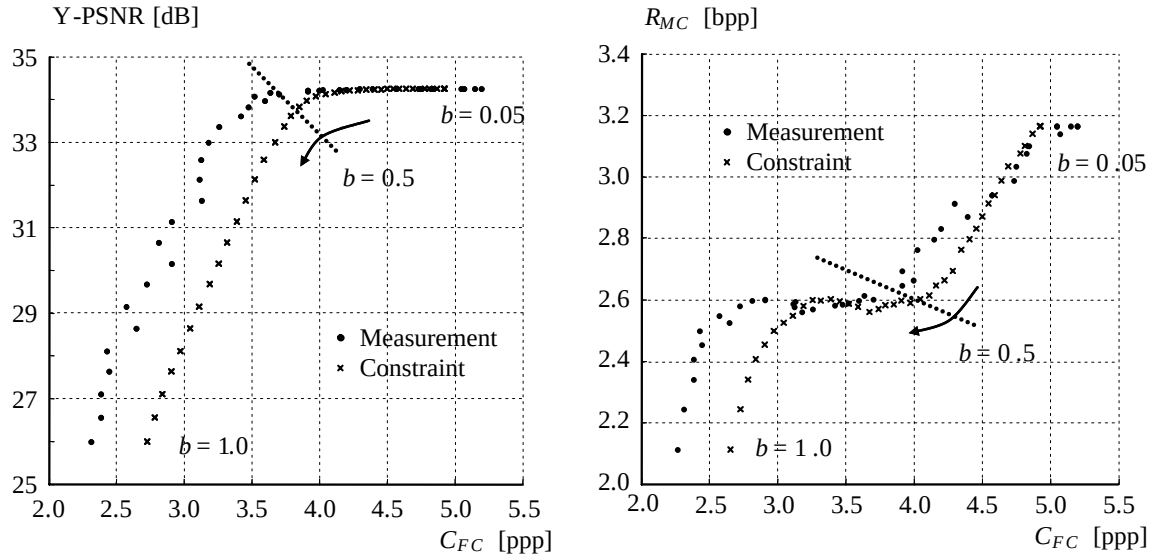

Figure 5.17: *(left) Decoding complexity vs. distortion plot for the decoding complexity constraint disparity compensation with α=0, β=0, γ=1, and N=10. b is calculated from the desired complexity constraint C_{FC} using (4.33) numerically. The disparity vector field is modified using a minimum MSE-criterion. Note that almost no degradation can be observed for b<0.5. (right) The corresponding rate vs. decoding complexity plot.*

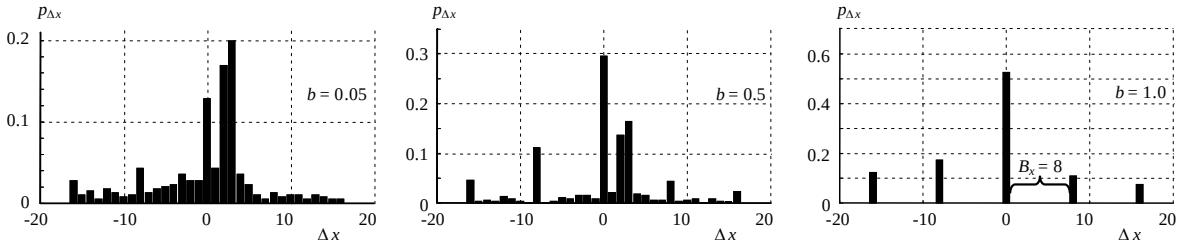

Figure 5.18: *Example distributions of the disparity vectors as a function of b and the block size B_x.*

a simple mapping from global encoding parameters (INTRA ratio α and SKIP ratio β) to the block mode decisions, and this also allows for global optimization instead of a mode selection on a block basis.

The separate examination of the initial delay and the mean delay reveals a significant difference in terms of overall system performance. For the initial delay, without any blocks already present in the client cache, a high decoding complexity and transmission data rate is observed which suggests a frequent use of the INTRA block mode. For the mean delay on the other side, it is observed that, provided that the user moves smoothly through the scene, rate-distortion optimal compression is preferable. This is mainly due to the fact that the decoding of dependent data actually resembles a data prefetching effect. RDTC optimal compression allows to adjust the trade-off between independently and dependently encoded parts of the input data.

Though not used in the real-time experiments, decoding complexity constrained disparity compensation can significantly reduce the decoding complexity with a slight decrease in coding efficiency. Especially for untextured regions this gives a significant gain also for the transmission data rate.

Comparative Evaluation The RDTC compression framework has been compared to other approaches in the former sections. The server centric approach considers conventional video coding of virtual views rendered at the server. While this turns out to be the most effective way in terms of a low decoding complexity and transmission data rate, the server has a high amount of work load to carry (decoding, rendering, and online compression). For multi-user scenarios this becomes infeasible.

The comparison to encoding with multiple representations shows that with a single scene description the mean delay can be very low when the RDTC compression framework is applied. The initial delay, however, is kept very small using encoding with multiple representations. This low worst case delay is at the expense of a significantly larger storage rate.

In [ZL05] and [SNC05] two systems have been introduced, both using hybrid video coding concepts for the compression of concentric mosaics. The former also implements a streaming system. However, the main difference is that RDTC optimized compression allows an adaptation to scenario specific constraints whereas in [ZL05, SNC05] fixed dependency structures are used. Compressed bitstreams provided by these techniques are conceptionally a subset of RDTC representations. Another difference is the block size chosen, which determines the least decodable unit. The mentioned systems use a block size of 16×16 which leads to significantly lower γ value than the system considered in this chapter. The decoding complexity and transmission data rate are proportional to $1/\gamma$. Therefore, the RDTC compression framework used in this chapter provides a better adaptation to a requested virtual view by transmitting and decoding fewer pixels that are not actually used for rendering.

5.9 Summary

In this chapter, a practical rate, distortion, transmission data rate, and decoding complexity (RDTC) optimization framework for the encoding and online streaming of image-based scene representations is discussed. To control the RD trade-off subject to TC constraints, trained models are proposed which allow for numerical global optimization for different target objectives. Both, optimization with respect to the initial delay in the beginning of a streaming session, and optimization with respect to the mean delay during online operation are investigated as well as joint optimization.

The optimization results are compared to common RD-optimization and independent encoding. The impact of human navigation decisions on the considered image-based rendering and streaming system during real-time operation is evaluated. Different modes of navigation are investigated and corresponding decoding complexity models are derived and incorporated into the final RDTC model. Further, a streaming testbed is implemented and used to measure the interactive streaming performance of RDTC optimized streams. Also, the impact of cache sizes on the overall performance is evaluated.

One main result is that despite the common understanding, the optimal way for encoding image-based scene representations becomes RD optimized from the second virtual view on (provided that the user moves smoothly through the scene). An explanation is that dependent blocks, transmitted and decoded during the request of the first virtual view, fill the cache with a high number of reference image blocks that can be reused for the second and subsequent virtual views. This is comparable to prefetching image data.

Real-time experiments show that an adaptation to both the client computational resources and the available transmission data rate, can be used to significantly decrease the initial user perceived delay compared to rate-distortion optimized compression. At the same time, a comparable mean delay during online operation is achieved while the storage rate increases only a little. Further, the cache size needed for optimal performance is lower for RDTC optimal compression compared to RD optimal encoding.

Compared to INTRA only encoding, the initial delay is slightly increased for RDTC representations while the mean delay is significantly lower. Also, the storage rate is much lower for RDTC optimal encoding compared to independent encoding at the same reconstruction quality.

The main contribution of this chapter is the practical RDTC optimization framework that is based on the theoretical findings of the previous chapter. The models derived for the RDTC system measures allow for a global optimization of a whole group of pictures. The significant differences in overall system performance when considering the initial delay and the mean delay, respectively, provide insights that can be used to design streaming systems using a more complex dependency structure than the one assumed in this chapter. This would allow us to achieve a higher compression efficiency.

6 Progressive rendering for RDTC optimized streams

In this chapter approaches for progressive transmission and rendering from compressed image-based scene representations are investigated. The compression scheme is based on the RDTC optimization framework for compression and interactive streaming as discussed in Chapters 4 and 5. Modifications to the compression framework that allow us to further adjust the online streaming behavior of the interactive streaming system are introduced. Based on this modified framework, a low quality reconstruction of a virtual view from only a fraction of the data actually needed for full quality view generation is evaluated. As further parts of the compressed representation are available at the client, the visual appearance of the virtual view is improved. The system response time and frame rate can be decreased when a degradation in quality is considered acceptable, e.g., during motion or at the beginning of a streaming session. Four different approaches for partial transmission and decoding of compressed data are discussed, and the system performance with respect to the user perceived delay and visual quality is evaluated. The system performance is also evaluated with respect to RDTC optimized streams, conventional rate-distortion optimization, and independent encoding of the input images.

Section 6.1 gives an overview of the objectives and techniques discussed in this chapter. In Section 6.1.1, the modifications to the compression procedure described in the previous chapters are introduced. Section 6.2 is dedicated to the progressive transmission, decoding and rendering schemes while Sections 6.3 and 6.4 discuss the results of this chapter and give a summary of the findings, respectively.

6.1 Overview

For the real-time experiments in the previous chapter it is assumed that upon a view request, the server assembles and transmits the relevant data for full resolution view generation. The client starts decoding as soon as data arrives, but, view interpolation is done after all relevant data to display a virtual view has arrived. However, it might be preferable to display a low quality approximation as soon as possible and then further refine the visual appearance when the system is idle (e.g., the user stops moving for a while). To achieve this, the transmitted bitstream has to be reordered. Principally, when access to the partial bitstream of every encoded pixel block is possible at the server (e.g., by using pointers to identify the beginning of the encoded representation of a block), an arbitrary transmission order of blocks needed for view generation is possible.

In common rate-distortion optimization, for interactive streaming of video or even image-based rendering data (here, the rate refers to the actual transmission data rate), the decision which parts of the data have to be sent, dependent on the network state, is done online. This requires computationally complex algorithms or a huge amount of side information (e.g., rate-distortion tables) available at run-time. The scheme in [RKG07] works on whole images and it has been shown that the complexity is considerably high, even in this simplified case

where the least decodable unit is a whole image rather than a single pixel block. Further, a deadline for the display of every virtual view is specified in that scheme. This requirement is replaced by a best effort approach in the system considered in this chapter, i.e., a virtual view is transmitted, decoded, and rendered using a small part of the data that allows to display a low resolution version. Until no other view request is received, the server continues with transmitting further data to improve the current virtual view until a full quality reconstruction is possible at the client. If a new view is requested before the previous one has been rendered in full quality, the new view is transmitted and decoded while the improvement of the previous view is abandoned. If the round trip time is large, much data for the previous view might still be transmitted at the time the progressive improvement has been already stopped by the client. Therefore, scheduling is suboptimal using this scheme, but is done with absolutely no computational overhead for the client and the server. The overhead data that is introduced with a large round trip time might be reduced by a user trajectory prediction at the server and with complex online scheduling approaches (as in [RKG07]). But, due to the dependencies between input images, most of the overhead data is likely to be relevant for neighboring virtual views.

Four different progression schemes are considered in this chapter. "Progressive interpolation", "viewport resampling", "image interleave", and "skip-scale progression" where the latter makes use of the implicit geometric information provided by the disparity information encoded with the representation.

In the following section, the modifications of the RDTC encoding scheme and the evaluation methodology for real-time experiments in the remainder of this chapter are introduced.

6.1.1 Multiple reference encoding

The major modification of the RDTC encoding framework introduced so far (compare to Section 5.1 on page 109) is that two possible block modes are added. Both of them use predictive coding as described in Section 4.1.1 on page 78, but, the reference is fixed to be the independently encoded image of a GOP (the anchor frame). The most obvious advantage of these ANCHOR-INTER and ANCHOR-SKIP modes is that the number of reference blocks is dramatically reduced on average as depicted in Figure 6.1 (compare to Figure 4.7 on page 88). In Figure 6.1, the requested block is encoded in ANCHOR-SKIP or ANCHOR-INTER mode. The disparity vector directly points to INTRA encoded blocks that can be used for prediction. The 12 white blocks do not have to be decoded as it is the case for the normal INTER and SKIP prediction block modes.

With this modification, multiple reference block encoding is performed as done, e.g., in [ZL05]. There, encoding on a frame basis is considered and the reference frame is only chosen from independently encoded frames rather than from arbitrarily encoded neighboring frames as in this chapter. I.e., a block can be predicted either from a neighboring frame or from the independently encoded anchor frame. The displacement vector for ANCHOR modes Δd_{anchor} (compare to Figure 6.1) is predicted from the distance to the independently encoded frame of a GOP and a mean disparity. This prediction is refined by the integer pel accurate displacement Δd that is actually stored with the bitstream:

$$\Delta d = \Delta d_{anchor} - \overline{\Delta d} \cdot \Delta f$$

The mean disparity $\overline{\Delta d}$ compensates for the overall mean displacement between the images

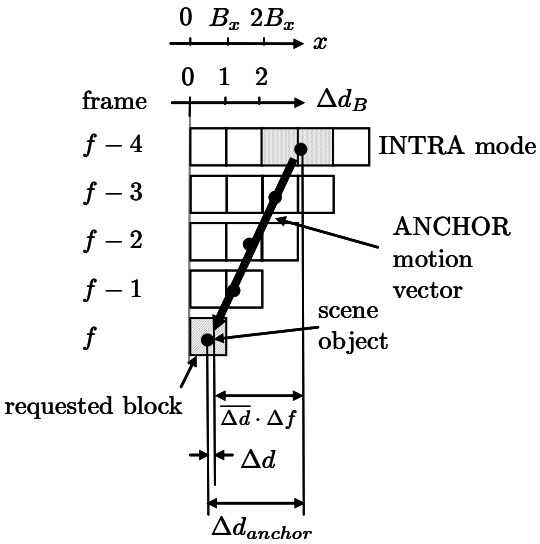

Figure 6.1: *A block encoded in ANCHOR mode is predicted directly from the independently encoded image at the beginning (or in the middle) of a GOP.*

of a structured scene representation. $\overline{\Delta d}$ can be calculated from the calibration data (e.g., the number of frames per full turn of the camera crane for concentric mosaics and an approximation of the focal length) assuming the scene resides at a constant depth (compare to Equation (A.7) in Appendix A.7 on page 165). Δf is obtained simply from the GOP size and the image number associated with the block request.

The mode selection using all five block modes (INTRA, INTER, SKIP, ANCHOR-INTER, and ANCHOR-SKIP) is done by modifying the methodology from the RDTC encoding framework (compare to Section 5.3 on page 114):

1. Block-based disparity estimation on the original images of a GOP is performed and a disparity vector field is produced.
2. The GOP is encoded for the six sample parameter settings $\boldsymbol{z}_i$=[α_i β_i q_i]^T with i=1...6, producing model fitting values for R, D, T, and C. According to $\boldsymbol{z}_i$ INTRA and INTER/SKIP blocks are distributed as follows:
   - INTRA mode encoding is chosen for the fraction α of blocks where the algorithm calculates the biggest MSE after disparity compensated prediction.
   - INTER mode encoding is selected for the fraction β of the remaining blocks introducing the biggest MSE after disparity compensated prediction.
   - Remaining blocks are encoded in SKIP mode.
3. An optimal parameter set $\boldsymbol{z}_{opt}$=[α_{opt} β_{opt} q_{opt}]^T is found using numerical (constrained) optimization on the trained model according to the objective function (e.g., (5.19)).
4. According to the optimal values α_{opt} and β_{opt}, INTRA blocks and INTER/SKIP blocks are distributed as described in step 2.
5. All INTER/SKIP mode representations are replaced by their corresponding ANCHOR representation if the MSE after disparity compensated prediction from the neighboring frame is greater than the MSE by prediction from the anchor frame.

This way, the better of the two prediction signals (either predicted from a neighboring or from the anchor frame) is used. The replacement of block modes does not decrease the rate-distortion performance of the encoded stream (but may improve it a little). The additional side information for signaling the choice of one out of five block modes compensates the gain by having a better prediction (due to the choice between two alternative references) during disparity compensation. Altogether 10%-20% of the SKIP and INTER modes are replaced for the two test sequences (see Appendix A.4 on page 162) which are compressed according to the RDTC optimization framework with a parameterization as described in Section 5.6 on page 125.

For data sets compressed using the modified representations, the decoding complexity according to the definitions in Chapter 4 (page 77) for a virtual view reconstructed at full quality is basically maintained. Though the progressive transmission and rendering techniques described in the remainder of this chapter can also be applied to a compressed representation without the modifications, the overall system performance is increased when using the modifications. The reason is that for progressive transmission the modified representations provides the advantage that a smaller subset of the data needed to reconstruct a virtual view can be assembled for a low quality reconstruction as discussed later in this chapter.

6.1.2 Evaluation methodology

Throughout this chapter, for every progressive transmission and rendering method, experimental results are given. These results are obtained mostly for three encoding schemes introduced in the former chapters:

- INTRA: All pixel-blocks are encoded using the INTRA mode,
- RD: Block mode decisions are made with respect to rate-distortion optimization adopted from common video coding, and
- RDTC: Optimized compression as described in the former Section 6.1.1

Compression using these schemes is carried out on GOP sizes of N=25 frames, a block size of B_x=B_y=8. The GOP structure is depicted in Figure 5.9. The two concentric mosaic data sets described in Appendix A.4 on page 162 are used (for the experiments they are both used in CIF resolution: N_x=352, N_y=288). The overall PSNR slightly differs for the three representations and the two datasets: The INTRA representation is encoded with R=1.0bpp at 40.3dB PSNR (40.9dB for the outdoor dataset). The rate-distortion optimized representation is encoded with 0.3bpp at 39.9dB (40.1dB) PSNR while the RDTC optimized representation is encoded with 0.39bpp at 40.0dB (40.3dB) PSNR (compare to Section 5.6 on page 125). Note that the independently encoded representation has a 2-3 times larger storage rate than the other representations at a comparable reconstruction quality. These rates include all information that is stored at the server (fixed length codewords on disparity vectors and Huffman coded and quantized transform coefficients).

For the evaluation in the remainder of this chapter, four motion scenarios are selected as depicted in Figure 6.2. There, in the upper left part the motion data of the virtual camera for the first 10 successive views of a streaming session is shown. No motion is performed and the cache is empty before the first view request. The initial delay and the behavior of the system for progressive refinement is studied using this trajectory. In the upper right of Figure 6.2 a trajectory consisting solely of rotation of the virtual camera is shown (43 view requests).

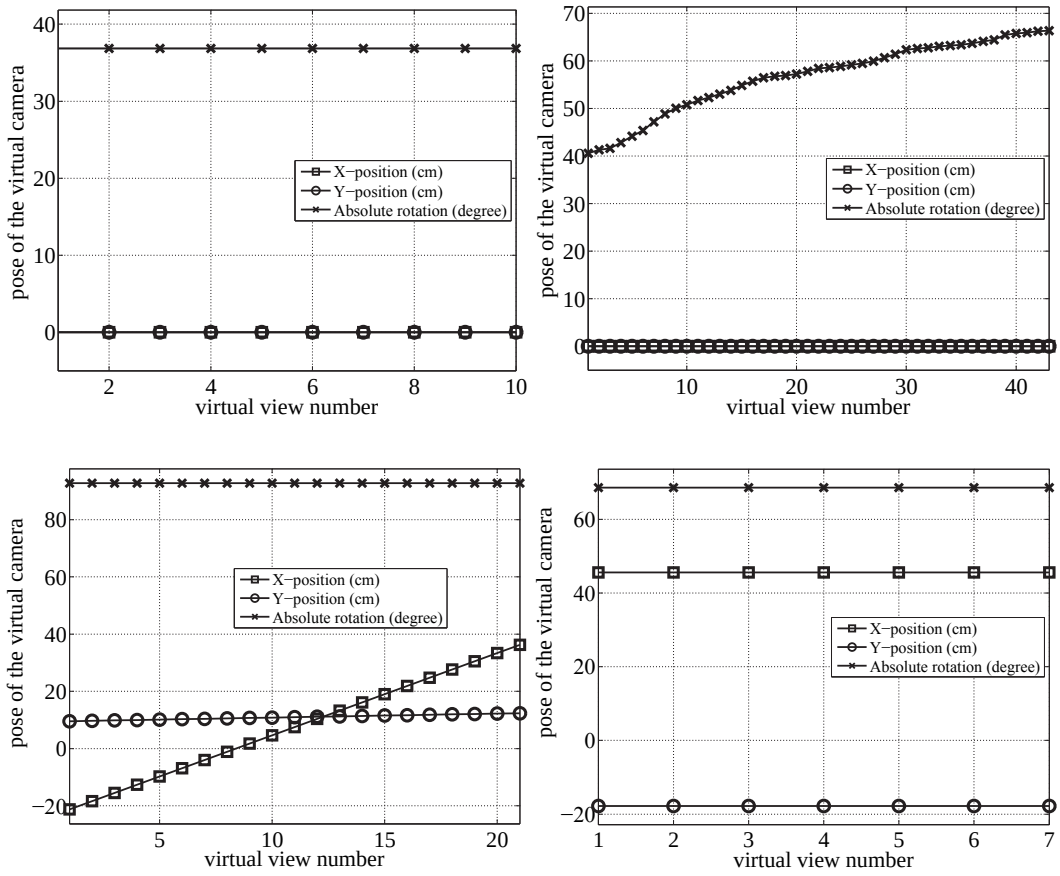

Figure 6.2: *The considered navigation scenarios shown as the motion data (translation and horizontal viewing angle on the X-Y plane) per view request.*

This trajectory is chosen when the user perceived delay during rotation of the virtual camera is to be measured whereas the lower left shows purely translational movement (21 view requests) used to determine the mean performance during translational movement of the user. The lower right part of Figure 6.2 shows a couple of views directly after translational movement (7 view requests). This trajectory is used to measure the system performance during standstill, i.e., the relaxation time until the full resolution version of the most recently requested view during translation can be displayed (similar to the initial delay but with a prefilled cache before the view request of the last view of the translation trajectory). Note that the actual timing of every view request depends on the delay its predecessor has introduced, thus, best effort transmission and rendering is performed. However, for all experiments conducted on rotation and translation of the virtual camera, either the computational power of the client device or the available transmission data rate is fully utilized. For the initial delay and standstill, these resources are fully utilized until the highest quality version of the virtual view is displayed.

In the real-time experiments, a cumulative delay trajectory (PSNR vs. user perceived delay) is used to quantify the performance of the streaming system using the initial delay and standstill evaluation patterns. For rotational and translational movement, the mean user perceived delay (the mean over all delays for all virtual view requests) versus the mean dis-

tortion (mean PSNR for all virtual view requests) is measured and plotted. Where shown, the PSNR is calculated with respect to rendering from uncompressed data at a resolution of 320×480 pixels (vertical×horizontal). All plots are produced by averaging over experiments using both data sets (two measurements for each data set, where the user trajectory is rotated by a fixed amount between the two measurements. This rotation is done manually to ensure that the virtual camera faces textured scene areas).

The system emulated by the testbed is a target system capable of decoding 1 million pixels per second (1Mpps) and a maximum channel bitrate of 1Mbps is chosen (identical to the system in Section 5.6 on page 125). The round trip delay is assumed to be negligible. The cache size is fixed at 5 MB for all experiments. The time for rendering (interpolation, perspective distortion, and depth correction according to [SH99]) is ignored (in the experiments rendering is carried out on the graphics hardware).

6.2 Progressive transmission, decoding, and rendering

In this section four progressive transmission, decoding and rendering approaches are investigated. Later, also the combination of these approaches is discussed.

6.2.1 Progressive interpolation

Virtual view generation consists of gathering samples of the plenoptic function (2.1) and interpolation of these samples at the pixel positions of the virtual view (compare to Section 2.4.1 on page 17). For real-time systems most often bilinear interpolation of intensity values of light rays hitting the scene geometry near the location where the light ray to be reconstructed intersects with the geometry is used (the camera proximity to the current light ray is also considered and should be low). Generally, to reconstruct one light ray, up to four blocks from the input data set have to be transmitted and decoded. But, an approximation of the light ray to be reconstructed can be made by nearest neighbor interpolation, reducing the maximum number of required blocks to only one. Upon a virtual view request, the server can first assemble the blocks containing the respective nearest captured light ray and their reference blocks. Then, the remaining information for full quality view generation using bilinear interpolation is assembled and transmitted. When the bitstream of the coarse approximation arrives at the client, it is decoded and rendered. The bitstream of the refinement is decoded while the approximation is being displayed to the user.

Additionally, for translational motion and rotation, an update of the virtual view is triggered only when the reconstruction time was below 50ms. I.e., the server assembles the bitstream but does not transmit the update without request from the client if the predicted decoding and transmission time (derived from the system settings as well as from the size and decoding complexity of the assembled bitstream) is exceeded. The additional computational overhead practically vanishes as only counters are used. This allows the system not to be blocked by updates from a previous view when a new view is triggered. Additionally, this allows the client to render the best possible view when, due to resolved dependencies, already enough information for rendering a better approximation is available. The delay and PSNR is then measured and accumulated for the best approximation that could be achieved.

This scheme is used also for the progression schemes described in the remainder of this chapter.

The "Progressive Interpolation" scheme reorders the bitstream of a virtual view into two quality layers (approximation and refinement). Though in the experiments only these two levels are considered, a finer granularity is possible in principle, e.g., by successively refining parts of the virtual view.

Experimental results

Figure 6.4 (left) shows experimental results for the initialization of a streaming session. The initial view request is triggered at time t=0 with preemptied cache. Then, in the real-time experiments, the time until the first (low quality) version of the requested view can be rendered is measured (response time). The distortion of the reconstructed virtual view is measured with respect to rendering from the uncompressed data using bilinear interpolation. Then, with update information arriving, the delay is accumulated and plotted. Markers denote the time when a virtual view update is rendered while lines connect successive view updates. The left most point in the curves is the initial delay, i.e., the time when the first approximation is displayed. Full quality is reached when the curve becomes horizontal. Schemes using progressive interpolation are denoted with an "L". The RDTC optimized scheme without progression is denoted as "RDTC" while RDTC optimization with progression is denoted as "RDTC L" and so on.

The overall degradation in quality for nearest neighbor interpolation compared to bilinear interpolation is about 5-6dB. This gap seems to be huge, but, it is hardly visible by a human observer during standstill as shown in Figure 6.3. During motion of the virtual camera nearest neighbor interpolation causes flickering.

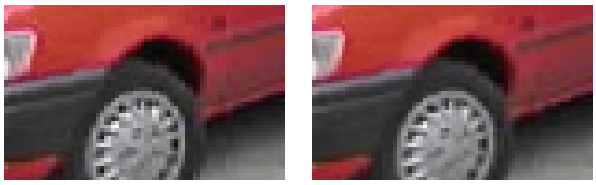

Figure 6.3: *A comparison of a virtual view rendered using nearest neighbor interpolation (left) and bilinear interpolation (right).*

The "RDTC L" scheme can display the first approximation after an initial delay of 550ms and catches up with full quality after another 130ms. Note that the initial delay for the RDTC optimized stream without the modifications discussed in Section 6.1.1 is at 580ms (not shown in the figures). The improvement of 30ms less delay for the modified version is due to the ANCHOR mode prediction modes. Similar gains are obtained for the schemes in the remainder of this chapter. 680ms after the view request, the full quality version of the view is rendered. Not surprisingly, this is also the initial delay for the RDTC scheme without progression. For the "INTRA L" scheme the first approximation can be displayed after approximately 500ms and catches up to full quality after another 480ms. This large gap compared to the "RDTC L" scheme is due to the fact that with dependent encoding, many of the blocks that are actually needed for bilinear interpolation are already present at the client

after the first approximation (though not used for interpolation). For the "RD L" scheme too much reference data that is actually not needed for rendering has to be transmitted and decoded resulting in a large initial delay of about 800ms. Without progression the initial delay is about 1000ms.

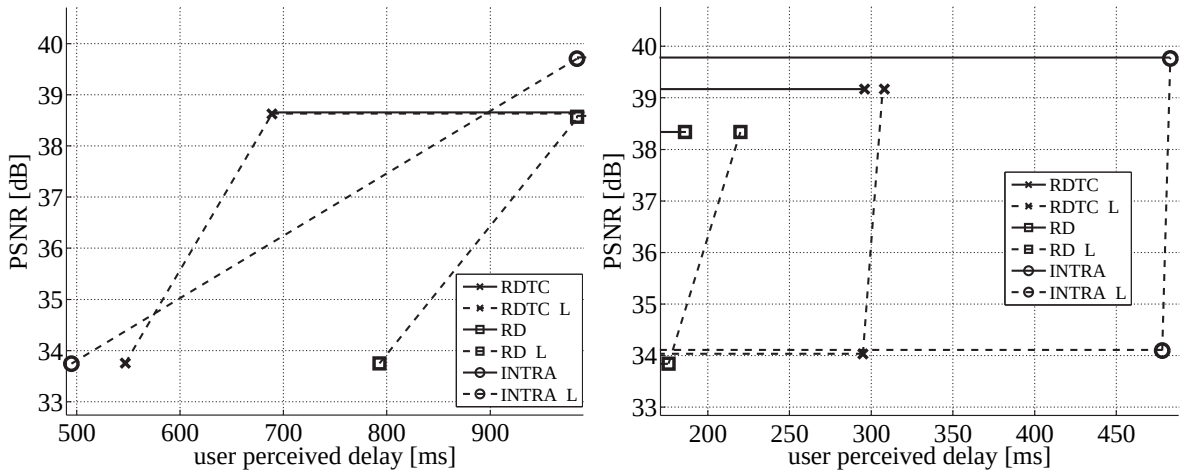

Figure 6.4: *The progressive refinement during the* ***initialization*** *(left) and* ***standstill*** *(right) for the* ***progressive interpolation*** *scheme.*

Figure 6.4 (right) shows the cumulative delay during standstill. Here, similar to the plot for the initial delay on the left, the time until versions of the last virtual view of a translation trajectory can be rendered, is evaluated. The curves start with a horizontal line segment indicating that the viewport is filled with the last virtual view. These line segments do not have accurate practical relevance but are added for clearer presentation and give an approximation from which PSNR of the previous view the refinement starts from.

The RD scheme without progression catches up earliest (180ms) to full quality. This is because most of the data needed for interpolation already has been transmitted for the last virtual view due to dependencies in the compressed representation. The corresponding scheme with progression "RD L" produces the final quality view after 220ms with an approximation rendered after 170ms. The "RDTC" scheme that principally needs less collateral blocks to be decoded, produces the first approximation after 290ms and catches up with full quality after another 20ms. The INTRA scheme shares almost no information between successive virtual views via the cache, and therefore, performs almost as worse as for initialization but catches up after 490ms. Note the steep slope (short time) between the approximation and the final results indicating that blocks cached for the previous virtual view are used for interpolation.

Figure 6.5 (left) shows the mean PSNR and mean user perceived delay per virtual view for rotation of the virtual camera. In this scenario, schemes based on rate-distortion optimization (RD) show a poor performance compared to the independently encoded and RDTC optimized representations. This is mainly due to the large amount of unneeded blocks that have to be transmitted and decoded for the RD schemes. Though the delay can be decreased by 50% (30%) for the INTRA (RDTC) scheme when using progressive interpolation compared to no progression, the degradation in (objective) quality is 4-5dB.

For translational motion as evaluated in Figure 6.5 (right), the performance of the "RDTC"

and "RDTC L" schemes are superior to both, INTRA and RD schemes with and without progression. Note that the rate-distortion optimized and independently encoded representations changed their place with respect to the rotation of the virtual camera and relative to the RDTC schemes. This is because for rotation less dependent blocks are advantageous whereas translational motion favors many dependent blocks. During transmission of many dependent blocks, as necessary for RD schemes, an effect similar to prefetching data is achieved (compare to Section 5.4 on page 119) which is not the case for the INTRA schemes. RDTC optimization balances this effect to some degree.

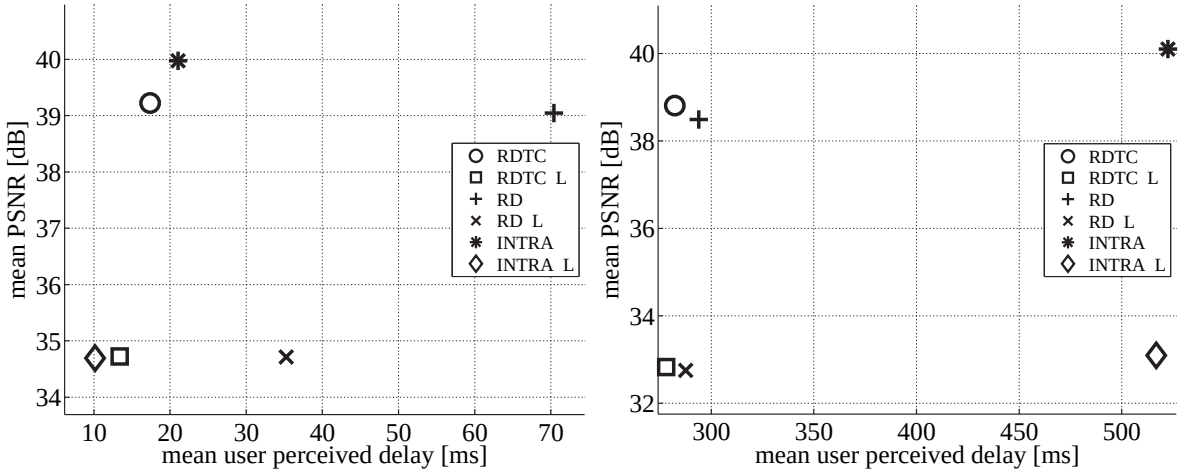

Figure 6.5: *The PSNR vs. mean user perceived delay for* ***rotation*** *(left) and* ***translation*** *(right) using* ***progressive interpolation****.*

6.2.2 Viewport resampling

The second progression scheme resamples the viewport (virtual view) to a lower resolution in order to reduce the number of pixels and therefore the number of blocks needed for rendering. For line light fields and concentric mosaics this means that the pixel columns for virtual view generation have a larger spacing than for full resolution rendering. Subsequently, the resulting low resolution view is upsampled to full resolution, again, using bilinear interpolation for display and distortion measurement. The viewport subsampling factor can be arbitrarily defined. In the real-time experiments an initial subsampling factor of eight is chosen. After the blocks needed for the low resolution version have been decoded, the subsampling factor is halved and the needed blocks for this more accurate view are assembled and transmitted. Then, again, the subsampling factor is halved, before the remaining blocks for the full resolution view are processed.

This viewport resampling scheme reorders the bitstream of a virtual view into four quality layers (three approximations and the final version). Though in the experiments only these four levels are considered, a finer granularity is possible in principle, e.g., by successively refining parts of the virtual view or decreasing the subsampling factor by one instead of halving. Also starting with a larger subsampling factor than eight is possible, but, results in low quality views. An example of the resulting visual artifacts is given in Figure 6.16. In the experiments the progression schemes using viewport resampling are denoted with a "C"

followed by the initial subsampling factor, e.g., "C8" for viewport resampling with an initial subsampling factor of 8.

Experimental results

Figure 6.6 (left) shows results for the initialization of a streaming session or subsequent virtual views that are very far apart (empty cache prior to the request). Viewport resampling achieves a lower initial delay than progressive interpolation (see Figure 6.4) at a significantly lower quality. With only 24dB PSNR, but a first approximation after 250ms the "INTRA C8" scheme is approximately 50% faster than progressive interpolation. The first update of the initial approximation after another 250ms achieves a PSNR of 28dB which is significantly lower than for progressive interpolation with 34dB PSNR at that delay. The latency for the second update and the final version equals as already all needed information is transmitted and the final view update is triggered due to the 50ms deadline. Similar findings can be made for the RDTC and RD schemes.

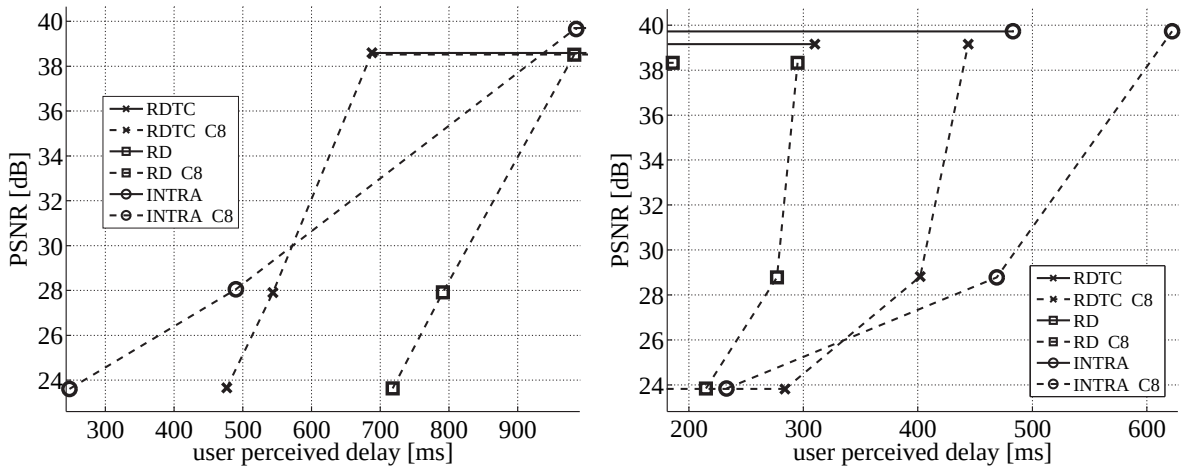

Figure 6.6: *The progressive refinement during the* ***initialization*** *(left) and* ***standstill*** *(right) for the* ***viewport resampling*** *scheme.*

Figure 6.6 (right) shows results for standstill after the translation trajectory. Generally, the schemes catch up with the full quality view much slower than for the progressive interpolation scheme.

The picture slightly changes with rotation of the virtual camera. As shown in Figure 6.7 (left), for the progressive "RDTC C8" scheme an increase of over 3dB PSNR compared to the corresponding progressive interpolation scheme can be observed. The same applies to the "INTRA C8" scheme whereas the delays for the RD schemes remain basically unchanged. The increased performance in quality comes with an increased mean delay. An explanation is that the viewport resampling schemes can update the view according to the 50ms deadline more often than the progressive interpolation scheme.

In Figure 6.7 (right), for translation of the virtual camera, the overall quality for the progressive schemes are low at approximately 22dB but a slight decrease of the user perceived delay compared to progressive interpolation can be observed. Viewport resampling achieves only

a very low objective quality. But, especially for rotation the blurring that is introduced gives the impression of motion blur which might be acceptable with the relatively low delay that is achieved, especially for the "RDTC C8" scheme.

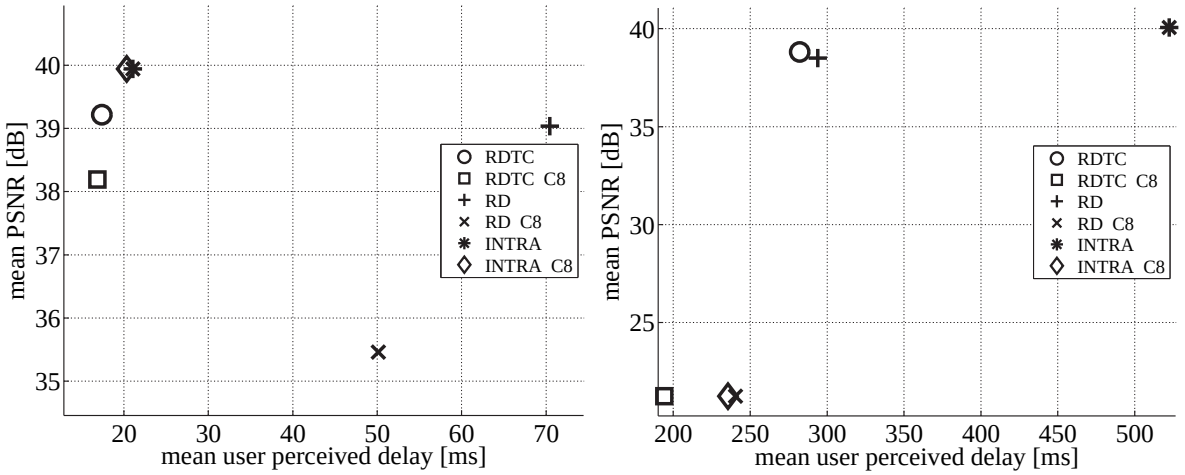

Figure 6.7: *The PSNR vs. mean user perceived delay for* ***rotation*** *(left) and* ***translation*** *(right) using* ***viewport resampling****.*

6.2.3 Image interleave

While the progressive interpolation scheme reduces the number of blocks that have to be transmitted by "subsampling" on the constant depth plane, the viewport resampling scheme subsamples the image plane of the virtual camera. Another possibility to progressively refine a virtual view is to resample the input images. I.e., virtually, an undersampled scene representation is assumed to be available, and the quality layers are assembled according to the virtual sampling factor of the input images. For line light fields and concentric mosaics this means that the input images have a larger spacing than for full resolution reconstruction. Blocks in images between the images that belong to the virtual data set have to be considered as reference blocks might have to be decoded there.

Again, the image interleave progression scheme reorders the bitstream of a virtual view into several layers. The subsampling factor can be arbitrarily defined, but, in the real-time experiments, subsampling factor sequences of [4,2,1] and [25,12,6,3,1] are used for coarse to fine approximation. The corresponding schemes are denoted with the suffix "I4" and "I25", respectively. Note that "I25" means that for the first approximation only the independently encoded frames are used as the GOP contains 25 images. After the blocks needed for the low resolution version are transmitted, the subsampling factor is adjusted according to the subsampling factor sequence and the needed blocks for the more accurate view are assembled. A finer granularity of the subsampling factor sequences as used in the experiments can be defined. An example of the resulting visual artifacts when using the image interleave progression scheme is given in Figure 6.16 on page 153.

Experimental results

Figure 6.8 (left) shows the cumulative delay versus distortion plot for the initialization of a streaming session. Obviously, with the progressive schemes, the full quality reconstruction

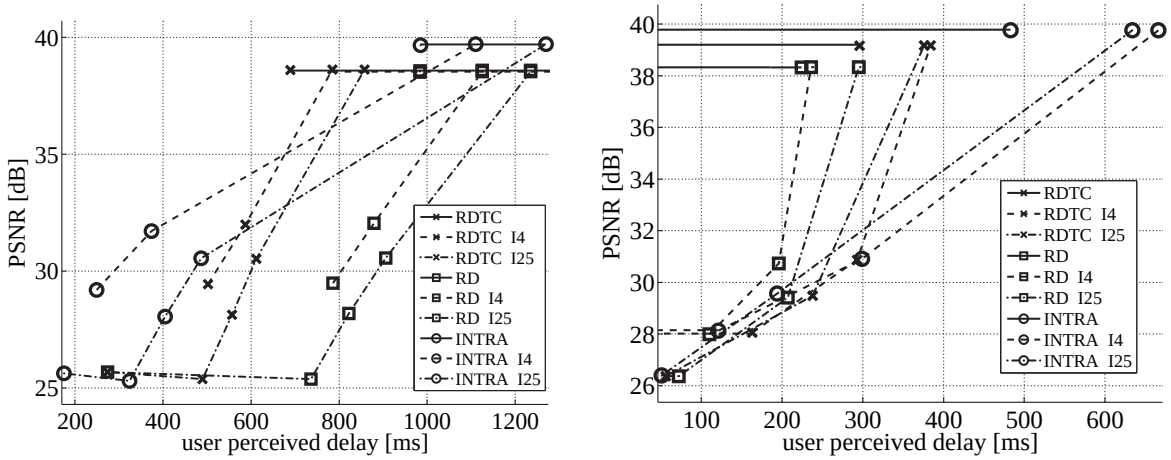

Figure 6.8: *The progressive refinement during the* ***initialization*** *(left) and* ***standstill*** *(right) for the* ***image interleave*** *progression scheme.*

of the virtual view takes longer than without progression. Strictly speaking, with image interleave progression, the bitstream for full quality view generation is not simply reordered by the server because the blocks for low resolution rendering might not be a true subset of the successive higher quality view. This means that more blocks have to be transmitted than needed for the full resolution version of the virtual view. There are two update steps for the "I4" and four for the "I25" scheme. The first view can be displayed for the "INTRA I25" scheme after less than 200ms. Though the quality is rather low, this is faster than with the other progression schemes. The "RDTC I25" as well as the "RD I25" schemes both display the first view approximation after 280ms. While "RDTC I25" catches up with full quality rendering after another approximately 550ms, the "RD I25" scheme achieves full quality after another 850ms, with the same cumulative delay as the "INTRA I25" scheme. The additional delay for the "I25" schemes until full quality is reached compared to the schemes without progression is about 200-300ms. The "I4" schemes do not achieve low delays, but, show 100-150ms less latency than the "I25" schemes for full reconstruction. The image interleave progression scheme is particularly suitable for INTRA representations as the number of transmitted blocks can be significantly reduced though a high delay for full quality views has to be considered.

In Figure 6.8 (right) the performance of image interleave progression is evaluated for standstill. The additional delay between using no progression and image interleave progression for full quality reconstruction is, again, considerable. This time, the reason is that for the progressive schemes fewer pixel blocks can be taken from the cache because the progression left some blocks untransmitted during the preceding translational motion. Nevertheless, first approximations can be rendered after a low delay of 60-80ms for all progressive schemes using the "I25" subsampling factor sequence at a low reconstruction quality of about 26dB. Full quality is reached a bit faster than for viewport resampling except for the INTRA scheme. Notably, the rate-distortion optimized representation (RD) reaches full quality rendering af-

ter approximately 220ms. This is mainly due to, again, the virtual prefetching effect this scheme induces.

The mean PSNR versus the mean user perceived delay is plotted in Figure 6.9 (left) for the rotation trajectory. INTRA and RDTC schemes show almost identical performance at 20ms delay and almost full quality. The RD schemes show more than three times longer latency due to many collateral blocks that are transmitted. Also, for translational motion, as shown in Figure 6.9 (right) RD schemes perform slightly worse compared to RDTC representations. Image interleave progression achieves quite low latency values with acceptable PSNR (e.g., 150ms at 29dB for "RDTC I4"). Visually, especially during translation, ghosting artifacts appear - as expected due to the undersampled virtual scene representation - that make rendering temporally inconsistent.

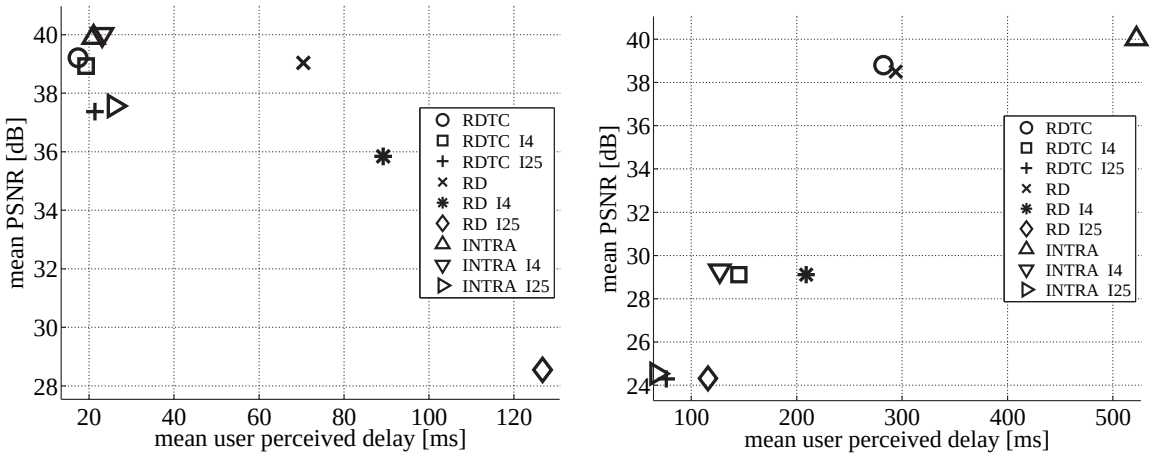

Figure 6.9: *The PSNR vs. mean user perceived delay for* ***rotation*** *(left) and* ***translation*** *(right) using the* ***image interleave*** *progression scheme.*

6.2.4 Skip-scale progression

The skip-scale progression scheme is closely related to the SKIP ratio. As discussed in the RDTC framework (introduced in Section 4.1.3 on page 80), the decoding complexity is related to β as shown in Figure 6.10 (compare to Equation (4.11) on page 92). The main idea is to generate a low quality approximation of the desired virtual view by only transmitting the independently encoded blocks and the disparity information of the blocks needed for rendering that are encoded in INTER and SKIP modes. As the residual error is not transmitted for the INTER block modes, error propagation during decoding does not allow to achieve the full quality reconstruction. But, the disparity information can be interpreted as implicit geometry information. This information is now used to shift the independently encoded and decoded blocks to their approximated position in the requested pixel blocks. In Figure 6.10 the path of cumulative decoding complexity is depicted. Starting from the lowest decoding complexity determined by the INTRA ratio α (at β=1), achieved when no residual error is decoded, the skip-scale progression scheme increases the decoding complexity by using more and more residual error information to decrease error propagation during decoding. The final complexity depends on the actual value of β the bitstream has been compressed

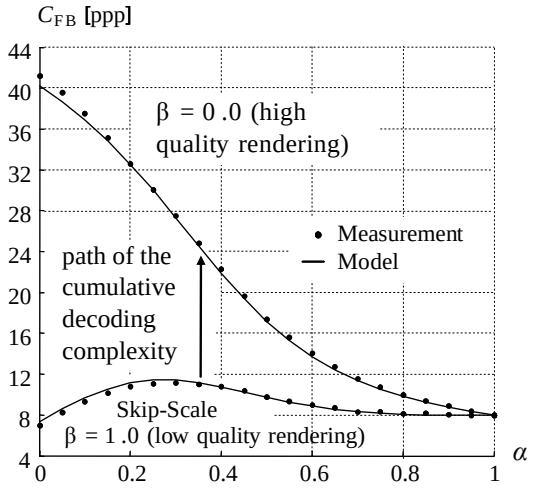

Figure 6.10: *The decoding complexity measured in pixel per pixel as a function of the INTRA ratio α and a GOP size of N=10 frames assuming full pel disparity compensation. The solid lines are calculated using (4.11) and are overlaid over their corresponding measurements.*

with (denoted as β=0 in Figure 6.10).

The skip-scale interleave progression scheme reorders the bitstream not on a block basis, but, INTRA blocks and disparity information are sent first. The client decodes the bitstream as if it only consisted of INTRA, SKIP, and ANCHOR-SKIP modes. Blocks that depend on non-INTRA blocks and are stored in the cache are marked as "infected" and the view is rendered. As the residual error arrives at the client, a new decoding pass is performed, now replacing the infected blocks for disparity compensation in the cache. INTRA blocks do not have to be decoded twice. The skip-scale progression scheme requires one decoding pass for all requested blocks for each update of the virtual view. Though a fine scalability is possible by partial transmission of the residual error, the present implementation only considers two quality levels. In the first level the geometric approximation of the virtual view is performed, in the update step the full quality view is reconstructed. An example of the resulting visual artifacts when using the skip-scale progression scheme is given in Figures 6.11 and 6.16. Due to error accumulation during tracking of the disparity information over a large number of images, mismatched blocks might appear in the approximated rendering.

Experimental results

In Figure 6.12 (left) results for initialization of a streaming session are given. The reconstruction quality is considerably high for the "RDTC Skip" scheme at a low latency compared to the other progression schemes at this quality. Again, a gap between the delay with and without progression for full resolution reconstruction is observed. The improvement for the "RD" scheme, e.g., compared to progressive interpolation, is even higher due to the larger number of INTER block modes. For the "INTRA Skip" scheme the skip-scale progression technique has no impact as no INTER or ANCHOR-INTER block modes are used there.

In Figure 6.12 (right) the impact of skip-scale progression during standstill is depicted.

Figure 6.11: *Typical visual artifacts for the skip-scale progression scheme. The blocks may be tracked over a distance of up to 13 images which leads to error propagation.*

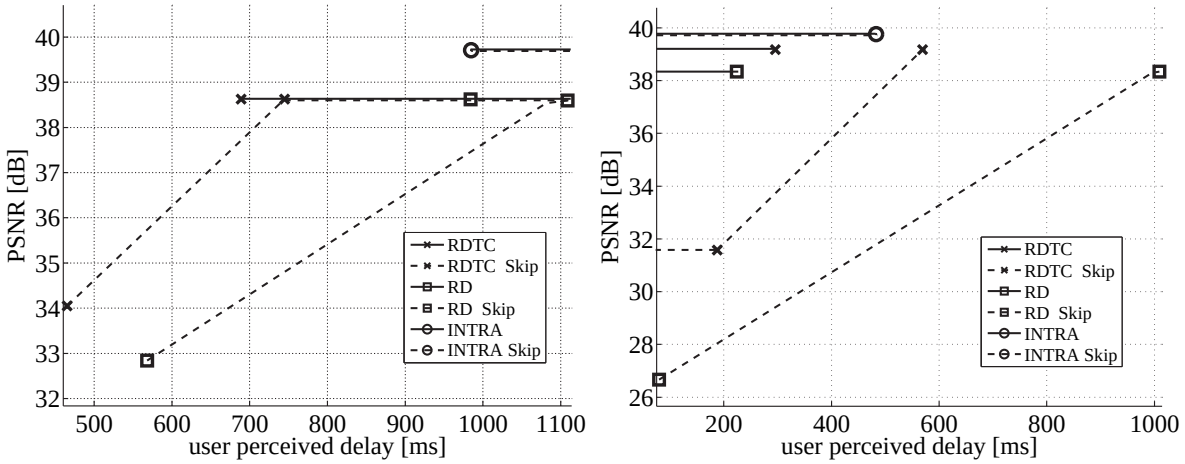

Figure 6.12: *The progressive refinement during the **initialization** (left) and **standstill** (right) for the **skip-scale** progression scheme.*

Again, the INTRA schemes show no difference for the progressive and the non progressive approaches. The "RDTC Skip" scheme catches up very late compared to other schemes. This is mainly due to the fact that a large number of infected blocks after translational movement have to be replaced. This is even worse for the "RD Skip" scheme, where almost the initial delay is measured until the full quality view can be displayed.

For rotation, the "RDTC Skip" scheme achieves low latency values with a relatively high PSNR while the "RD Skip" scheme shows a significantly higher delay. Compared to other progression schemes this scheme benefits from the skip-scale progression as depicted in Figure 6.13. For translational motion, both the "RDTC Skip" and "RD Skip" schemes show the highest PSNR at a low latency of approximately 120-140ms compared to the other three progression schemes. On the INTRA schemes skip-scale progression has no impact.

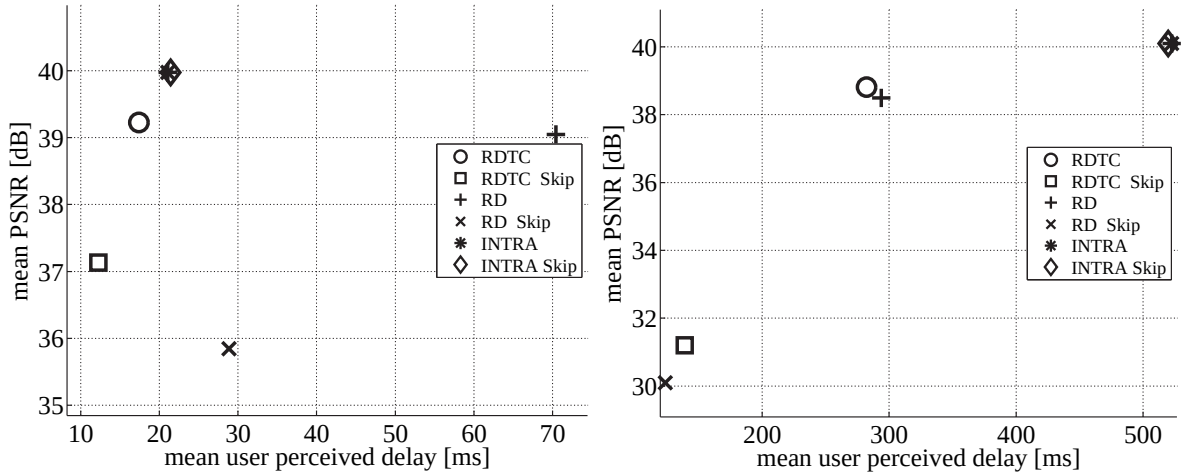

Figure 6.13: *The PSNR vs. mean user perceived delay for* ***rotation*** *(left) and* ***translation*** *(right) using the* ***skip-scale*** *progression scheme.*

6.2.5 Combining the progression schemes

As the progression schemes introduced in the former sections work in completely different domains (on the interpolation, the image plane of the virtual view, the image sampling grid, and the disparity and residual information), in principle they can be combined. This combination also involves choosing a suitable progression scheme for the mode of navigation that the user chooses. E.g., for an RDTC optimized representation, the first virtual view during initialization can be rendered using image interleave progression (lowest latency of 280ms) while during the same session, for rotation of the virtual camera, skip-scale progression might be the optimal choice (mean user perceived delay of 12ms at 37dB PSNR). However, the introduced progression schemes can also be used simultaneously. "C8" and "I25" schemes are not considered in the following as they produce very low quality renderings that are particularly too blurry or temporally inconsistent. In the former sections of this chapter it has been shown that, in general, the "RDTC" scheme achieves the best distortion versus delay trade-off when no progression is performed. Therefore, this representation is chosen as compression scheme for experiments in this section.

Experimental results

Figure 6.14 (left) shows results for the initialization of a streaming session using different combinations of progression schemes. The initial delay without progression is approximately 680ms. The corresponding skip-scale scheme "RDTC Skip" needs about 60 ms to achieve the same PSNR of about 38.7dB with an initial delay of 460ms at 34dB PSNR. The scheme with viewport resampling and progressive interpolation combined "RDTC C4 L" only achieves an initial delay of 540ms at low 28 dB. The delay to full reconstruction is identical to the scheme without progression at 680ms. Image interleave progression combined with skip-scale and progressive interpolation "RDTC I4 Skip L" can display the first virtual view after 380ms at a PSNR of 29dB. The same scheme, but, without progressive interpolation achieves practically the same results. These values are better than for any single version

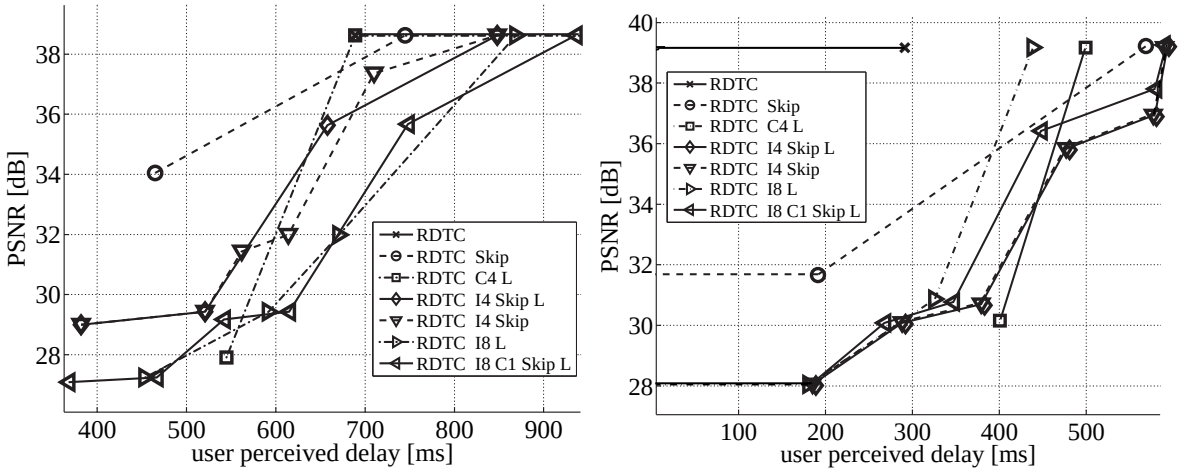

Figure 6.14: *The progressive refinement during the* ***initialization*** *(left) and* ***standstill*** *(right) for* ***combined*** *progression schemes.*

of the progression schemes at this quality. Image interleave progression together with progressive interpolation "RDTC I8 L" and additionally with skip-scale progression "RDTCI8 C1 Skip L" cannot compete with the other combined schemes during initialization.

In the right of Figure 6.14, the behavior with respect to standstill is shown. As expected, the RDTC scheme without progression renders the full reconstructed version of the virtual view earliest. All other schemes need over 400ms to catchup. The "RDTC Skip" scheme renders an approximation after 200ms at a PSNR of 31.5dB PSNR, but with a delay of over 560ms for full quality reconstruction. All other schemes show a worse performance.

For rotation of the virtual camera, again, the "RDTC Skip" scheme performs at the lowest delay of 12ms at 37dB PSNR as shown in Figure 6.15 (left). The corresponding scheme without progression has a mean delay of 18ms at 39.5dB PSNR. Actually, all other schemes perform worse than these two in the mean delay versus mean PSNR sense.

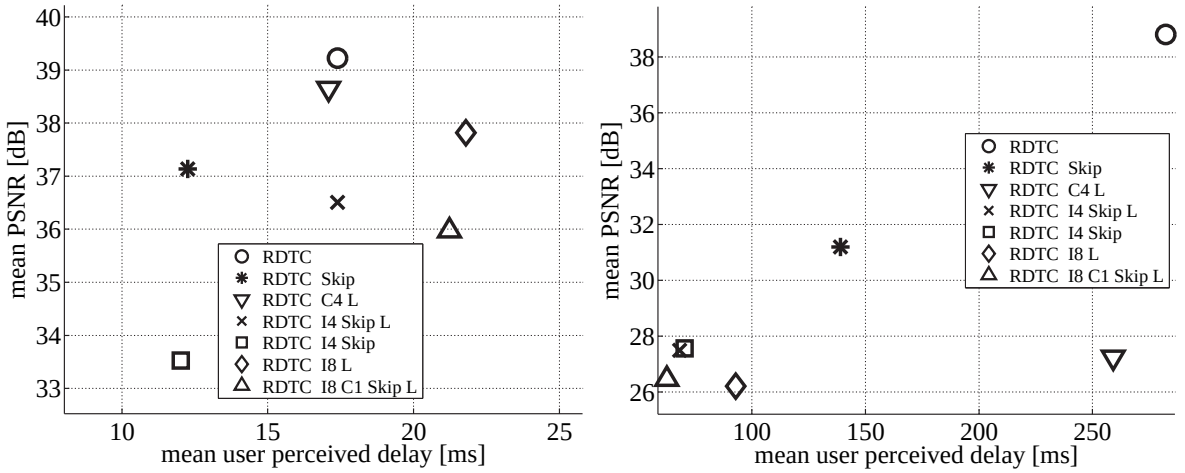

Figure 6.15: *The PSNR vs. mean user perceived delay for* ***rotation*** *(left) and* ***translation*** *(right) using* ***combined*** *progression schemes.*

The performance for translational movement is shown in Figure 6.15 (right). Here, if a low delay is preferred for the application, the "RDTC I8 C1 Skip L" scheme combining all of the four progression approaches performs at a latency of only 50ms with a PSNR of only 26.5dB. A more reasonable PSNR value is achieved by the "RDTC Skip" scheme at 140ms mean delay and 31dB PSNR.

Effectively, during motion of the virtual camera and for a reasonable PSNR level, the "RDTC Skip" scheme has a delay of 30-50% of the scheme without progression (which in turn performs considerably better than RD and INTRA). For the initialization and standstill patterns, "RDTC Skip" can display the first approximation in 70% of the time the scheme without progression needs while the time for full reconstruction is about 10% (for initialization) and 50% (for standstill) larger.

6.3 Discussion

Overall, the RDTC compression framework in conjunction with the skip-scale progression scheme allows for the best performance in the delay versus quality sense using the introduced progression schemes. Different kinds of visual artifacts appear for the different progression techniques as shown in Figure 6.16.

In the literature, specifying a deadline for the presentation of a virtual view is proposed (e.g., [RKG07]). This makes sense when the main goal is to provide a certain response time (and a certain frame rate), but requires online scheduling. The simple scheduling approach as introduced in the former sections, namely, the truncation of the transmission of a view upon a new virtual view request and updating when the expected delay is below a threshold (50ms in the experiments), can already provide interactive rates for remote walkthrough applications with practically no computational overhead.

The main drawback of the presented evaluation is that the round trip delay is ignored. However, online scheduling is expected to give another significant gain at the cost of a high computational load either for the server or the client and can be build on top of the introduced schemes. The same applies to user motion prediction and data prefetching which allow to handle significant round trip delays. Other schemes introduced for scalable video coding like fine granular scalability (FGS) schemes can also be set on top of the discussed progressive transmission schemes.

Schemes based on wavelet decomposition (e.g., [LWLZ02b, PS01]) generally allow progressive transmission and decoding. However, the computational complexity is higher than with hybrid video coding concepts as used in this chapter. Further, with disparity compensated prediction, the same issues as discussed with INTRA, RDTC, and RD optimized representations have to be addressed for interactive streaming. Therefore, also for schemes based on wavelet decompositions, the insights of this chapter may be useful.

6.4 Summary

In this chapter four techniques for progressive transmission, decoding, and rendering have been investigated in detail. The schemes are designed to allow for a consistent quality distribution within the virtual view, i.e., no black areas where no information is available, are

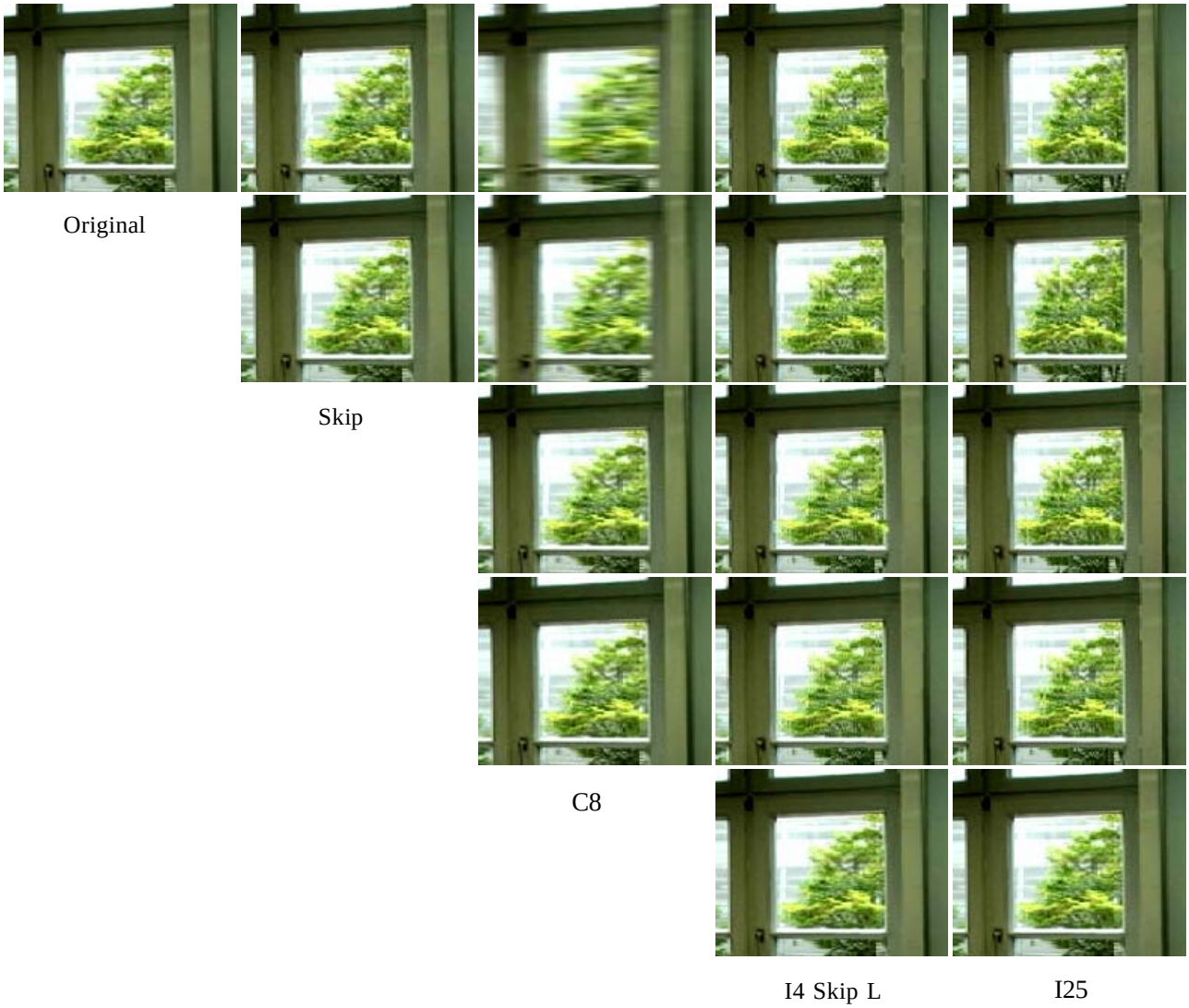

Figure 6.16: *Visual artifacts during initialization (empty cache prior to the view request) for different progression schemes on the RDTC optimized representation (contrast enhanced for clearer presentation). The view generated from uncompressed data is shown in the left. Note that the schemes in the Figure catch up with the original after different numbers of updates (from top to bottom). Blurring, aliasing and misplacement artifacts are visible.*

allowed, as often used in practice (e.g., [ZL05, SNC05]). "Progressive interpolation" allows us to display a low resolution version of the requested virtual view by replacing the bilinear interpolation during rendering with a simpler nearest neighbor interpolation. This can reduce the number of pixel blocks that have to be transmitted by up to 50% for INTRA encoded data. Progression using the "viewport resampling" method is carried out by subsampling the image plane of the virtual camera. In this way, the number of requested blocks can be arbitrarily reduced, but, for dependently encoded scene representations, the reduction in transmission data rate and decoding complexity is limited for reasonable quality levels. The "image interleave" progression scheme subsamples the input images to a virtually subsam-

pled data set. The client requests blocks only from a subset of the original images. With dependently encoded data, this scheme can achieve low latency values while the quality can become quite low (e.g., 100ms at 24dB PSNR for RDTC optimized streams and translational motion). "Skip-scale progression" uses the disparity information to approximate a virtual view without decoding the residual error of predicted blocks. In the refinement passes the residual error is incorporated again to allow full quality rendering without error propagation during decoding at the client.

The investigated progression schemes can be used to efficiently reduce the user perceived delay for interactive streaming of compressed scene representations. Especially, for image-based data sets compressed using the RDTC compression framework, progressive transmission provides a way to flexibly adapt the transmitted bitstream to the user navigation decision without the need for computational complex online scheduling performed by the server. The real-time experiments show that interactive rates are possible for realistic scenarios (e.g., frame rates of 10Hz for the most difficult translational motion at a quality level of 40dB with a maximum channel bit rate of 1Mbps and a computational power of 1Mpps). By using different progression schemes for different modes of navigation (initialization, rotation, translation) during a remote navigation session, the system can adapt to the needed quality versus user perceived delay trade-off without any computational overhead at the server or client.

The main contribution of this chapter is the introduction and evaluation of the "skip-scale" progression scheme. Results show that performing a view approximation using the implicit geometry information given by the disparity information gives good results in terms of the quality versus delay trade-off. Another contribution is the detailed analysis of progressive transmission, decoding and rendering with respect to rate-distortion optimization, RDTC optimization, and independent encoding.

7 Conclusion

In this thesis the acquisition and rendering of unstructured image-based scene representations using a multi-sensor platform and the compression for interactive streaming of densely sampled image-based scene representations is investigated.

TRIVIS

The considered multi-sensor platform TRIVIS consists of three video cameras and a laser range finder. The device is to be moved manually in front of a scene to acquire images and laser scans that are registered in space and time to form an unstructured image-based scene representation. The main features of TRIVIS and the introduces algorithms are:

- a full metric device calibration by jointly considering the depth sensor and three cameras,
- a robust pose estimation procedure for long image sequences under random motion of the acquisition device,
- a sensor data fusion algorithm for 3D scene reconstruction, and
- a robust real-time rendering procedure using multiple local models with outlier detection and removal.

The calibration procedure considers all sensors jointly and provides a metric reconstruction of the physical device setup and all intrinsic sensor parameters. The accuracy of the proposed algorithm is comparable to state-of-the-art multi-camera calibration techniques, but additionally provides the scale of the reconstructed parameters and also provides the calibration of the laser scanner. The pose estimation algorithm is designed for long image sequences with random motion of the hand-held acquisition device. Overall, fewer image correspondences than for the single camera case are needed to obtain an accurate reconstruction due to the precalibration of the multi-sensor platform. Further, mismatches are less likely due to guided matching. Experiments show that the data fusion algorithm for 3D scene reconstruction provides better per pixel depth maps than single sensor techniques. Further experiments show that the outlier detection and removal algorithm based on the images and their local geometry allows for a better overall reconstruction quality than conventional view reconstruction techniques in the presence of noise in the depth maps.

RDTC optimization

In the second system that is investigated in this thesis interactive streaming of densely sampled image-based scene representations is considered. The classical rate-distortion optimization approach using hybrid video coding concepts is extended to a trade-off between the storage rate, distortion, transmission data rate, and the decoding complexity. A theoretical model for such an RDTC space with a focus on the decoding complexity and, in addition,

the impact of client side caching on the RDTC measures is considered and evaluated. Experimental results qualitatively match those predicted by the theoretical models and show that an adaptation of the encoding process to scenario specific parameters like computational power of the receiver and channel throughput can significantly reduce the user perceived delay or required storage for RDTC optimized streams compared to RD optimized or independently encoded scene representations.

Beside the theoretical evaluation, a practical RDTC optimization framework is discussed. To control the rate-distortion trade-off subject to TC constraints, trained models are proposed which allow for numerical global optimization with different target objectives. Both optimization with respect to the initial delay in the beginning of a streaming session and optimization with respect to the mean delay during online operation is investigated as well as joint optimization. One main result is that despite the common understanding, the optimal way for encoding image-based scene representations becomes RD optimized from the second virtual view on if the user moves smoothly through the scene. The main conclusions from the theoretical analysis are approved and the impact of finite size caching is evaluated using a real-time streaming testbed.

Further, progressive transmission of image-based scene representations compressed using hybrid video coding concepts is investigated. Four progression schemes are introduced and evaluated: Progressive interpolation, viewport resampling, skip-scale progression and image interleave progression. While the schemes can be easily combined with any other progression scheme that is common for video coding (e.g., fine granular scalability - FGS), they can also be used simultaneously. The system response time can be significantly reduced by sacrificing the reconstruction quality for an initial approximation of the virtual view. The Skip-Scale progression scheme is based on an implicit geometry reconstruction using the motion information provided by the disparity compensation side information from the compressed bitstream.

Outlook

The techniques introduced and evaluated in this thesis have been built on top of state-of-the-art techniques. However, there are many issues that have to be addressed in future work.

For the acquisition and rendering process using a multi-sensor platform it is desirable to have a full automatic self-calibration process. The method introduced in this thesis relies on manually chosen camera-laser sensor point correspondences. The main reason here is that the laser scanner does not produce overlapping scans which makes the correspondence search between subsequent or even wider spaced scans practically impossible. However, linking the depth estimation and pose estimation stages might make it possible to find a remedy. The use of more complex features like line or curve segments would make the pose estimation even more reliable especially in man made environments and with only little textured scene objects. The incorporation of all images and laser scans to produce more reliable depth maps would also improve the rendering results. To make use of the very flexible acquisition procedure, an online view planning algorithm would be very useful to decrease the needed number of input images to a minimum.

The main drawback of the RDTC optimization framework as introduced in this thesis is the restriction of group of pictures to a sequence of images that have been approximately cap-

tured on a line. The extension of the decoding complexity models for hierarchical and 2D dependency structures would allow a significantly higher coding performance while providing control over all four RDTC measures.

For progressive transmission, the use of the proposed schemes in addition with other techniques that provide scalability would be of interest. Further, the investigation of online packetization and scheduling approaches in conjunction with RDTC optimization would allow to adapt to the user behavior and changes in the channel throughput at run-time.

A Appendix

A.1 Feature extraction and region matching

To determine the spatial relationship between images taken from previously unknown positions, the first step is to extract a sufficiently large number of image correspondences. The number of correspondences needed for the subsequent algorithms to produce reliable results depends on, e.g., constraints on the camera's motion trajectory or the number of free intrinsic parameters. To establish feature correspondences, usually, selected features (interesting and distinguishable points, lines, etc.) are extracted in each image. Then, the extracted features are matched between two or more images by considering the local neighborhood of the features.

In this work, point correspondences are considered and are extracted using the Harris corner detector [HS88] and are matched by maximizing the zero-mean normalized cross correlation (ZNCC) on a 5×5 pixel window around the feature points. The similarity measure ZNCC is defined as

$$ZNCC = \frac{\sum_{\boldsymbol{W}} \left(J(\boldsymbol{L}(x,y)) - \overline{J}\right) \cdot \left(I(x,y) - \overline{I}\right) \cdot w(x,y)dxdy}{\sqrt{\sum_{\boldsymbol{W}} \left(J(\boldsymbol{L}(x,y)) - \overline{J}\right)^2 \cdot w(x,y)dxdy} \cdot \sqrt{\sum_{\boldsymbol{W}} \left(I(x,y) - \overline{I}\right)^2 \cdot w(x,y)dxdy}} \tag{A.1}$$

on a window $\boldsymbol{W}$ in image I and a corresponding region $\boldsymbol{L}(\boldsymbol{W})$ in image J. $w(x,y)$ is a weighting function that, e.g., if it is chosen to be a Gaussian, weights differences near the region borders less. Usually, this weighting function is set to $w(x,y)$=1 to speed up the matching process. The mean intensity in the pixel regions in image I is:

$$\overline{I} = \frac{1}{|\boldsymbol{W}|} \sum_{\boldsymbol{W}} I(x,y)dxdy$$

and in image J:

$$\overline{J} = \frac{1}{|\boldsymbol{W}|} \sum_{\boldsymbol{W}} J(\boldsymbol{L}(x,y))dxdy.$$

Matching of point features using the ZNCC is done automatically for pose estimation as described in Section 3.4 on page 50.

Image regions can also be compared using the sum-of-squared-differences (SSD). The dissimilarity between two image regions in images I and J is given as

$$SSD = \sum_{\boldsymbol{W}} \left(J(\boldsymbol{L}(x,y)) - I(x,y)\right)^2 \cdot w(x,y)dxdy. \tag{A.2}$$

The SSD measure is sensitive to changes in the light conditions between the images under consideration. In this work, matching for pose estimation and calibration is done using the

ZNCC to be as much as possible independent from lighting conditions. The SSD measure is used for scene reconstruction as here non-Lambertian surface properties, such as specularity and subsurface scattering, and lighting changes during image acquisition are an integral property of the desired (image-based) model and therefore should be included in the modeling process.

A.2 Establishing correspondences for multi-sensor calibration

Point correspondences between the images of two or more cameras for joint calibration of the multi-sensor platform as described in Section 3.2 on page 35 are established automatically. As a laser pointer is used in a darkened environment, there is only one point of interest for every image triplet that is captured. This makes it very easy to match this single feature point between the camera images. Figure A.1 shows a magnification of the projection of the laser spot in the first of the three cameras of TRIVIS. The rest of the images is basically black (or static and therefore can be separated from the foreground), and the spot is easily detected using a simple intensity threshold. As the image of the laser spot covers many pixels and as sub pixel accuracy is desired for calibration, a 2D Gaussian is fitted [SMP05].

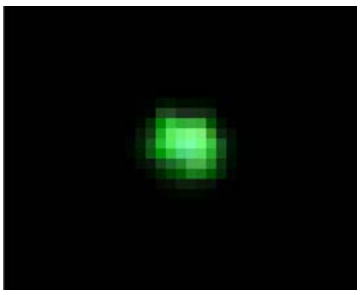

Figure A.1: *The image of the laser spot in the first camera (magnification of a cutout)*

A depth scan that is obtained by the laser scanner is illustrated in Figure A.2. If the pointer device crosses the line of sight of the scanner, a peak can be detected manually as marked with a circle in Figure A.2. The corresponding images of the laser spot are obtained via the synchronization of the acquired images and the laser scans in time.

A.3 Bundle adjustment

Bundle adjustment is a geometric parameter estimation problem. "Bundles" are two or more light rays leaving a scene point and propagating toward several camera centers. The cameras' intrinsic and extrinsic parameters, as well as the geometric structure of the captured scene are "adjusted" jointly in order to minimize the squared Mahalanobis distance of the estimation error

$$\|\boldsymbol{\epsilon}\|^2_{\Sigma} = \|\boldsymbol{f}(\boldsymbol{p}) - \boldsymbol{m}\|^2_{\Sigma}. \tag{A.3}$$

Here, the vector valued function $\boldsymbol{f}$ defines the predefined model and $\boldsymbol{p}$ is the parameter vector that is to be estimated. For computer vision tasks these parameters are usually the

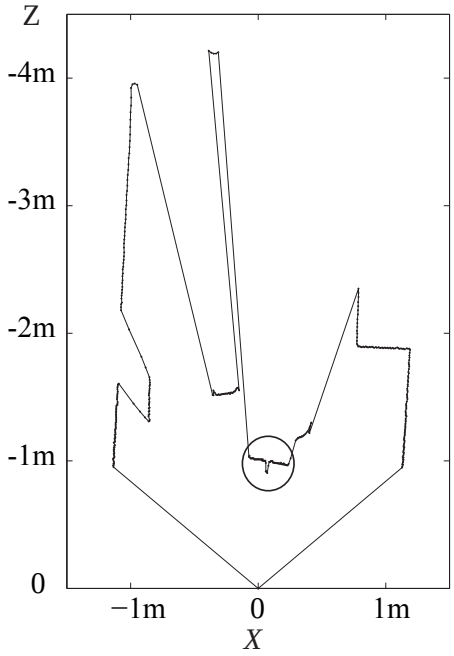

Figure A.2: *One scan containing the laser pointer used for calibration.*

intrinsic and extrinsic camera parameters and the 3D feature point locations. The distance between the approximated measurement vector $\hat{\boldsymbol{m}} = f(\boldsymbol{p})$ and the given measurement vector $\boldsymbol{m}$ is to be minimized in the least squares sense considering the (estimated) variance of the measurement error (indicated by the subscript Σ). The measurement vector $\boldsymbol{m}$ consists of image coordinates of the projections of the same scene point in several images.

In order to minimize (A.3), most often the Levenberg-Marquardt algorithm is used [Lev44, Mar63]. This requires to determine partial derivatives of $\boldsymbol{f}$ with respect to the parameters. This can be done in two ways. The first way is to find analytical expressions of these partial derivatives as it is done for the joint calibration procedure in Section 3.2 on page 35 and for the common task of pose estimation in structure from motion problems in Section 3.4.4 on page 53. For problems where a closed form solution can not be found or this solution is unstable for any reason, the partial derivatives can be determined using numerical methods as it is done in 3.4.5 on page 55. Both methods heavily rely on accurate starting points in order to converge in a global optimum.

In this dissertation the implementation in [LA04] is used for all the optimization procedures that involve bundle adjustment. There, the sparsity of the overall problem is exploited to increase the computational efficiency during solving the large scale problems that have to set up with many hundreds of images and several thousands of image correspondences. Implementation details, accuracy evaluations, as well as a discussion about the convergence behavior and stability are given in [TMHF99, LA04, HZ04, Sch07].

A.4 Test data and error measures

The test datasets "classroom" and "courtyard" used in the experiments in Chapters 4, 5 and 6 consist of normal concentric mosaics captured with a camera radius of 1.5 meters using the setup shown in Figure A.3. 1525 frames at CIF resolution with a field of view of approximately 40 degrees are recorded. Example images are shown in Figure A.4. The test sets are critically sampled according to section 2.3.1 on page 13. Rendering is performed by bilinear interpolation using the four nearest light rays from the two cameras nearest to the considered light ray intersecting at the constant depth. Figure A.5 shows example renderings from the test sets at 40dB reconstruction peak signal to noise ratio (PSNR). The reconstruction PSNR is calculated on the luminance component as:

$$PSNR = 10 \cdot log_{10} \left(\frac{255^2}{MSE} \right) [dB] \tag{A.4}$$

where the mean squared error MSE is calculated as:

$$MSE = \frac{1}{\|I\|} \cdot \sum_{p \in I} \left(I(p) - \hat{I}(p) \right)^2 \tag{A.5}$$

Here, p is a pixel position in the original image or pixel block I and the reconstructed (decompressed) image or block $\hat{I}$, containing $\|I\| = \|\hat{I}\|$ pixels.

Some of the results in Chapter 4 use the signal to noise ratio as the distortion measure which is defined as

$$SNR = 10 \cdot log_{10} \left(\frac{P_S}{P_N} \right) [dB]. \tag{A.6}$$

Here, P_S and P_N are the power of the signal and the background noise, respectively.

Figure A.3: *Camera crane for the acquisition of concentric mosaics.*

Figure A.4: *Representative reference images captured using a concentric mosaics setup. An indoor "classroom" (top row) and outdoor "courtyard" scene (bottom row).*

A.5 Evaluation methodology for streaming of structured representations

Experimental results for the theoretical models in chapters 4 and 5 are produced in the following way unless otherwise noted:

1. With the system parameters fixed, a set of N consecutive frames (one GOP) is randomly chosen from the densely sampled line light fields (concentric mosaics in 4CIF and CIF resolution, indoor and outdoor scenes).
2. Disparity estimation is performed on the original image data producing a disparity field with one disparity value per block ($s = 1$).
3. If not stated explicitly, b is calculated using (4.8).
4. According to α and β, blocks are marked randomly as INTRA or INTER/SKIP blocks.
5. A large number of random access experiments is performed for a specific access pattern. Either a cache is simulated or not. INTRA, INTER, and SKIP block requests are counted.
6. The theoretical model is evaluated using the actual system and signal parameters.
7. Steps one to six are repeated for a few hundred times.
8. The mean of the results from steps 5 and 6 are compared and plotted.

A.6 Sampling of light fields

Figure A.6 shows the capture geometry for a 2D lightfield. The camera plane is simplified to a line t while the focal plane with distance f_c to the camera line is simplified to the line v. A light ray L is to be reconstructed for the display by the virtual camera at C_V. A suitable ray acquired by the nearest capturing camera at C_R serves as approximation. The maximum distance z_{max} and minimum distance z_{min} between possible scene points and the capturing line t along the ray L is assumed to be known. The objective is to minimize the maximum

Figure A.5: *Example renderings from the two test data sets using a single constant depth assumption.*

of the spatial error Δv between the true projection of the (unknown) scene point and its approximation in the reference image captured from C_R. The optimal ray K intersects v half way between v_{max} and v_{min}. The maximum spatial error becomes $e_{max} = (v_{min} - v_{max})/2$. As can be seen from the figure, K and L intersect at the optimal depth z_{opt}.

On the right side of Figure A.6 a sampling spectrum is shown with most densely packed spectrum when choosing ΔX_{max} according to Equation (2.3) where d is chosen to be the pixel width on v. With the optimal choice of ΔX_{max}, the maximum spatial error e_{max} in the captured image becomes one pixel. The corresponding spectrum is bounded by the two lines according to the minimum and maximum depth in the scene. The replica due to discrete sampling touch the base band spectrum of width ΔB.

From Figure A.6 (right) it also can be seen that a simple box filter as indicated by the dashed lines would cause heavy aliasing during reconstruction while the skewed box filter indicated by the dashed-dotted lines can reconstruct the signal correctly by linear interpolation of a few rays at z_{opt}. The derivation of the spectra and filter design can be found in [CCST00].

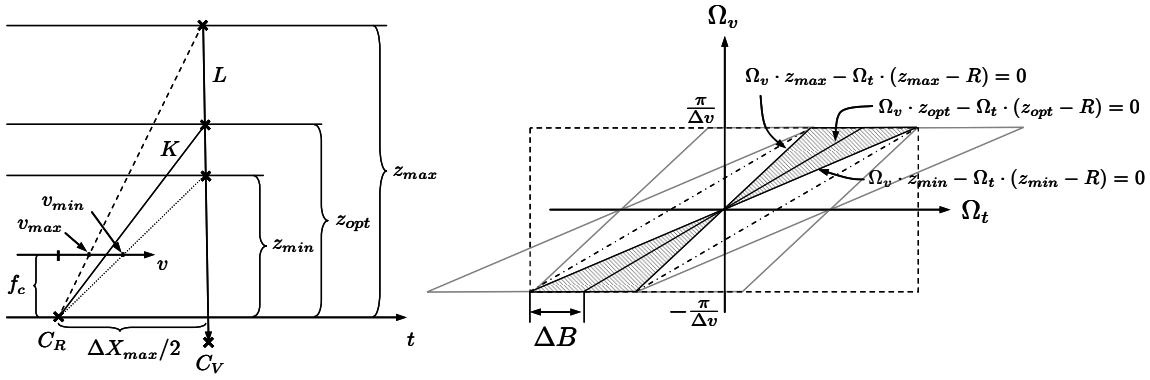

Figure A.6: *Capture geometry for a 2D light field (left) and an exemplary spectrum (right).*

A.7 Sampling of concentric mosaics

Figure A.7 (left) shows the 2D acquisition geometry for concentric mosaics (top view). A camera moves along the circle with radius R. At an angle β the camera captures an image. The captured light rays are registered by the tuple (α, β). The angular disparity $\Delta\alpha$ for an object at distance r is approximated by:

$$\Delta\alpha = -\frac{\Delta\beta \cdot r}{r - R}. \tag{A.7}$$

To minimize the maximum angular error when choosing a light ray assuming the scene object at distance z_{opt} instead of the unknown true depth, the minimum and maximum disparities are averaged:

$$\overline{\Delta\alpha} = -\frac{\Delta\beta}{2} \cdot \left(\frac{z_{max}}{z_{max} - R} + \frac{z_{min}}{z_{min} - R} \right). \tag{A.8}$$

The corresponding optimal depth z_{opt} can be determined from:

$$-\frac{\Delta\beta \cdot z_{opt}}{z_{opt} - R} = \overline{\Delta\alpha} \tag{A.9}$$

Which gives:

$$z_{opt} = R \cdot \left(1 - 2 \cdot \left(\frac{z_{min}}{z_{min} - R} + \frac{z_{max}}{z_{max} - R} \right)^{-1} \right)^{-1}. \tag{A.10}$$

The Nyquist frequency (e.g., [Nyq28]) for sampling images on the camera path can be determined graphically from the spectral support of the concentric mosaic light rays indexed by (α, β). The slopes of the spectral lines for an object at distance r to the center of the concentric mosaic is the inverse disparity from Equation (A.7) and satisfies [ZC03b]:

$$\Omega_\alpha \cdot r - \Omega_\beta \cdot (r - R) = 0. \tag{A.11}$$

As the depth is assumed to be bounded, the maximum bandwidth ΔB in Ω_β direction can be determined at $\Omega_\alpha = \frac{\pi}{\Delta\alpha}$ by

$$\Delta B = \frac{\pi}{2 \cdot \Delta\alpha} \cdot \left(\frac{z_{min}}{z_{min} - R} - \frac{z_{max}}{z_{max} - R} \right). \tag{A.12}$$

With $\Delta B = \frac{2 \cdot \pi}{\Delta \beta}$ and the approximation $\Delta \alpha \approx \frac{\Delta u}{f_c}$ for real cameras with a limited field of view and a focal length of f_c the minimum angular spacing $\Delta \xi_{max} = \frac{1}{2} \cdot \Delta \beta$ of two subsequent images captured on the camera path becomes

$$\Delta \xi_{max} = \frac{2 \cdot \Delta u}{f_c} \cdot \left(\frac{z_{min}}{z_{min} - R} - \frac{z_{max}}{z_{max} - R} \right)^{-1}. \tag{A.13}$$

Finally, if the desired output resolution is equal to the input resolution of the capturing cameras then $\Delta u = d$ where d is the pixel diameter. Equation (2.5) can be used to determine the minimum angular spacing for sampling images using a concentric mosaic set up.

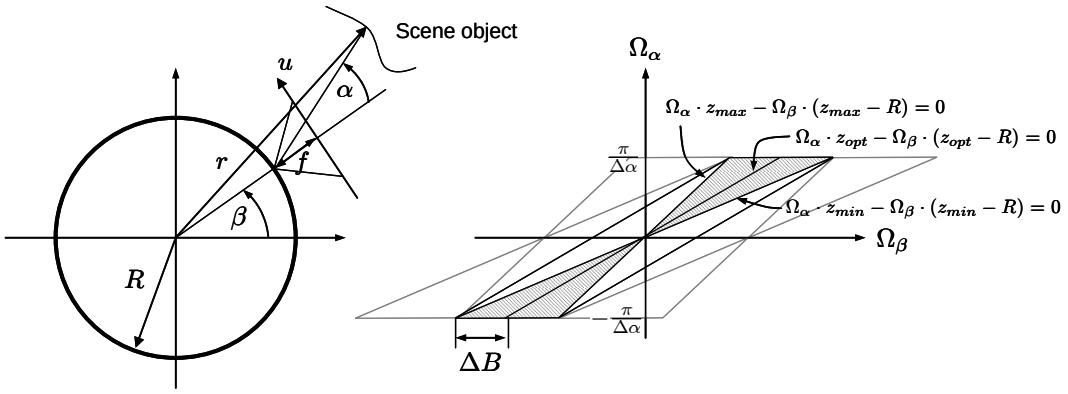

Figure A.7: *Concentric mosaic acquisition geometry (left) and rectangular sampling spectrum (right).*

A.8 The impact of the single reference block ratio on the decoding complexity

To decrease the number of axes in the plots in Chapters 4 and 5, the influence of the single reference block ratio b is averaged out for different random access experiments. The actual value of b has an impact on the overall complexity as it is shown in Figure A.8 for the decoding complexity C_{FB} from Equation (4.33) in Section 4.2.3 on page 90. However, most of the disparity field implementations have values of b that can be approximated using Equation (4.8). Modifying b by choosing suboptimal motion vectors (in the sense of minimization of the mean squared error) can significantly change the overall decoding complexity as shown in Section 5.7 on page 129. The decoding complexity in the example in Figure A.8 can vary by a factor of 7 depending on the value of b and α.

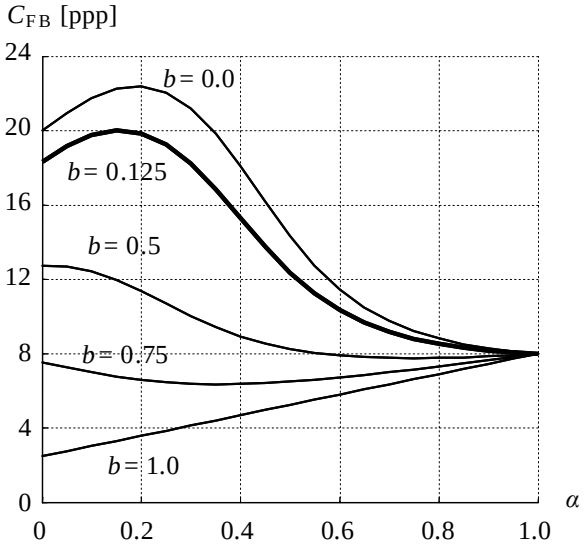

Figure A.8: *The decoding complexity C_{FB} from Equation (4.33) measured in decoded pixel per rendered pixel as a function of the INTRA ratio α and the single reference ratio b. The SKIP-rate is $\beta = 0.75$ for a GOP size of $N = 13$ frames assuming full pel motion compensation on pixel blocks of size $B_x = B_y = 8$ and with γ=1/8.*

Bibliography

Publications by the author

[BCS05] Ingo Bauermann, Subhasis Chaudhuri, and Eckehard Steinbach. Sensor data fusion for the acquisition and compression of RGBZ concentric mosaic. *Workshop on Immersive Communication and Broadcast Systems*, 2005.

[BMS04] Ingo Bauermann, Matthias Mielke, and Eckehard Steinbach. H.264 based coding of omnidirectional video. *International Conference on Computer Vision and Graphics*, 2004.

[BMS08] Ingo Bauermann, Werner Maier, and Eckehard Steinbach. Progressive rendering from RDTC optimized streams. *IEEE International Conference on Multimedia and Expo*, 2008.

[BPS06a] Ingo Bauermann, Yang Peng, and Eckehard Steinbach. RDTC optimized streaming for remote browsing in image-based scene representations. *Third International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.

[BPS06b] Ingo Bauermann, Yang Peng, and Eckehard Steinbach. Receiver- and channel-adaptive compression for remote browsing of image-based scene representations. *Picture Coding Symposium*, 2006.

[BS03] Ingo Bauermann and Eckehard Steinbach. An image-based rendering framework incorporating multiple representations. *Vision, Modeling And Visualization*, 2003.

[BS04a] Ingo Bauermann and Eckehard Steinbach. Further lossless compression of JPEG-images. *Picture Coding Symposium*, 2004.

[BS04b] Ingo Bauermann and Eckehard Steinbach. Low-complexity image-based 3D gaming. *Vision, Modeling And Visualization*, 2004.

[BS05] Ingo Bauermann and Eckehard Steinbach. Joint calibration of a range and visual sensor for the acquisition of RGBZ concentric mosaics. *Vision, Modeling And Visualization*, 2005.

[BS06a] Ingo Bauermann and Eckehard Steinbach. Analysis of the decoding complexity of compressed image-based scene representations using hybrid video coding concepts. (LKN-MTG-TR-BAUERMANN-06-01), 2006.

[BS06b] Ingo Bauermann and Eckehard Steinbach. Receiver- and channel-adaptive compression and streaming for image-based scene representations using hybrid video coding concepts. (LKN-MTG-TR-BAUERMANN-06-02), 2006.

[BS07a] Ingo Bauermann and Eckehard Steinbach. Analysis of the decoding complexity of compressed image-based scene representations. *International Conference on Image Processing*, 2007.

[BS07b] Ingo Bauermann and Eckehard Steinbach. Encoding parameter estimation for

RDTC optimized compression and streaming of image-based scene representations. *International Conference on Image Processing*, 2007.

[BS07c] Ingo Bauermann and Eckehard Steinbach. A theoretical analysis of the RDTC space. *Packet Video Workshop*, 2007.

[BS08a] Ingo Bauermann and Eckehard Steinbach. RDTC optimized compression of image-based scene representations (Part I): A theoretical analysis. *IEEE Transactions on Image Processing*, 2008.

[BS08b] Ingo Bauermann and Eckehard Steinbach. RDTC optimized compression of image-based scene representations (Part II): Practical coding. *IEEE Transactions on Image Processing*, 2008.

[SBS08] Florian Schweiger, Ingo Bauermann, and Eckehard Steinbach. Joint calibration of a camera triplet and a laser rangefinder. *IEEE International Conference on Multimedia and Expo*, 2008.

[VBS04] Eswar Kalyan Vutukuri, Ingo Bauermann, and Eckehard Steinbach. Decoding complexity-contrained rate-distortion optimization for the compression of concentric mosaics. *Picture Coding Symposium*, 2004.

General publications

[AB91] E. H. Adelson and J. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 1991.

[AC01] D. G. Aliaga and I. Carlbom. Plenoptic stitching: A scalable method for reconstructing 3D interactive walkthroughs. *SIGGRAPH '01*, 2001.

[AFYC03] D. Aliaga, T. Funkhouser, D. Yanovsky, and I. Carlbom. Sea of images. *IEEE Computer Graphics and Applications*, 2003.

[ARG04] A. Aaron, P. Ramanathan, and B. Girod. Wyner-Ziv coding of light fields for random access. *IEEE International Workshop on Multimedia Signal Processing*, 2004.

[BAT03] H. Baltzakis, A. Argyros, and P. Trahanias. Fusion of laser and visual data for robot motion planning and collision avoidance. *Machine Vision and Application*, 2003.

[BBM+01] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. *SIGGRAPH '01*, 2001.

[BC00] A. Broadhurst and R. Cipolla. A statistical consistency check for the space carving algorithm. *11th British Machine Vision Conference*, 2000.

[Ber71] T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. 1971.

[BS05] A. Bartoli and P. Sturm. Structure from motion using lines: Representation, triangulation and bundle adjustment. *Computer Vision and Image Understanding*, 2005.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986.

[CBCG02] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. *SIGGRAPH '02*, 2002.

[CBL99] C.-F. Chang, G. Bishop, and A. Lastra. LDI tree: A hierarchical representation for image-based rendering. *SIGGRAPH '99*, 1999.

[CCST00] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong. Plenoptic sampling. *SIGGRAPH '00*, 2000.

[CF77] C. H. Smith Chen, W. H. and S. C. Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on Communications*, 1977.

[CG04] C.-L. Chang and B. Girod. Receiver-based rate-distortion optimized interactive streaming for scalable bitstreams of light fields. *IEEE International Conference on Multimedia and Expo*, 2004.

[Che95] S. E. Chen. Quicktime VR – an image based approach to virtual environment navigation. *SIGGRAPH '95*, 1995.

[CKS00] J. Chai, S.-B. Kang, and H.-Y. Shum. Rendering with non-uniform approximate concentric mosaics. *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, 2000.

[CL96] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. *SIGGRAPH '96*, 1996.

[CLF98] E. Camahort, A. Lerios, and D. Fussell. Uniformly sampled light fields. *Eurographics Workshop on Rendering*, 1998.

[CLM07] G. Chen, Y. Liu, and N. Max. Real-time view synthesis from a sparse set of views. *Image Communication*, 2007.

[CM02] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[CM06] P. A. Chou and Z. Miao. Rate-distortion optimized streaming of packetized media. *IEEE Transactions on Multimedia*, 2006.

[Col96] R. T. Collins. A space-sweep approach to true multi-image matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[CS02] P. A. Chou and A. Seghal. Rate-distortion optimized receiver-driven streaming over best-effort networks. *Packet Video Workshop*, 2002.

[CTMS03] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *SIGGRAPH '03*, 2003.

[CW93] S. E. Chen and L. Williams. View interpolation for image synthesis. *SIGGRAPH '93*, 1993.

[CZRG06] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod. Light field compression using disparity-compensated lifting and shape adaptation. *IEEE Transactions on Image Processing*, 2006.

[DBY98] P. E. Debevec, G. Borshukov, and Y. Yu. Efficient view-dependent image-based rendering with projective texture-mapping. *Eurographics Rendering Workshop*, 1998.

[Dia03] P. Dias. Three dimensional reconstruction of real world scenes using laser and intensity data. *PhD thesis, University of Aveiro*, 2003.

[DL03] J. Duan and J. Li. Compression of the layered depth image. *IEEE Transactions on Image Processing*, 2003.

[DM97] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. *SIGGRAPH '97*, 1997.

[DMMV05] M. Do, D. Marchand-Maillet, and M. Vetterli. On the bandlimitedness of the plenoptic function. *IEEE International Conference on Image Processing*, 2005.

[DTM96] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. *SIGGRAPH '96*, 1996.

[DTM+98] P.E. Debevec, C.J. Taylor, J. Malik, G. Levin, G.Borshukov, and Y. Yu. Image-based modeling and rendering of architecture with interactive photogrammetry and view-dependent texture mapping. *Symposium on Circuits and Systems*, 1998.

[EHBR98] S. El-Hakim, C. Brenner, and G. Roth. An approach to creating virtual environments using range and texture. *ISPRS Commission V Symposium*, 1998.

[ESG99] P. Eisert, E. Steinbach, and B. Girod. Multi-hypothesis, volumetric reconstruction of 3D objects from multiple calibrated camera views. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[ESG00] P. Eisert, E. Steinbach, and B. Girod. Automatic reconstruction of stationary 3D objects from multiple uncalibrated camera views. *IEEE Transactions on Circuits and Systems for Video Technology: Special Issue on 3D Video Technology*, 2000.

[ESK03] J.-F. Evers-Senne and R. Koch. Image based interactive rendering with view dependent geometry. *Eurographics 2003*, 2003.

[ESK05] J.-F. Evers-Senne and R. Koch. Image-based rendering of complex scenes from a multi-camera rig. *Vision, Image and Signal Processing*, 2005.

[ESNK06] J.-F. Evers-Senne, A. Niemann, and R. Koch. Visual reconstruction using geometry guided photo consistency. *Vision, Modeling, and Visualization*, 2006.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[FCOL00] F. Fleishman, D. Cohen-Or, and D. Lischinski. Automatic camera placement for image-based modeling. *Computer Graphics Forum*, 2000.

[FH04] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[FKK04] J.M. Frahm, K. Koser, and R. Koch. Pose estimation for multi-camera systems. *Symposium für Mustererkennung*, 2004.

[FLB06] J.-S. Franco, M. Lapierre, and E. Boyer. Visual shapes of silhouette sets. *3rd International Symposium on 3D Data Processing, Visualization and Transmission*, 2006.

[FWZ03] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. *International Conference on Computer Vision*, 2003.

[FZ02] C. Früh and A. Zakhor. Data processing algorithms for generating textured

3D building facade meshes from laser scans and camera images. *International Symposium on 3D Processing*, 2002.

[GD99] C. Geyer and K. Daniilidis. Catadioptric camera calibration. *International Conference on Computer Vision*, 1999.

[GEM+99] B. Girod, P. Eisert, M. Magnor, E. Steinbach, and T. Wiegand. 3D image models and compression - synthetic hybrid or natural fit ? *International Conference on Image Processing*, 1999.

[GG91] A. Gersho and R.M. Gray. Vector quantization and signal compression. *Kluwer Academic Publishers*, 1991.

[GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. *SIGGRAPH '96*, 1996.

[Gir87] B. Girod. The efficiency of motion-compensating prediction for hybrid coding of video sequences. *IEEE Journal of Selected Areas in Communications*, 1987.

[Gir93] B. Girod. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications*, 1993.

[Gir94] B. Girod. Rate-constrained motion estimation. *SPIE Conference on Visual Communication and Image Processing*, 1994.

[Gir00] B. Girod. Efficiency analysis of multihypothesis motion-compensated prediction for video coding. *IEEE Transactions on Image Processing*, 2000.

[Gra84] R. M. Gray. Vector quantization. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1984.

[HCTL02] Y.-P. Hung, C.-S. Chen, Y.-P. Tsai, and S.-W. Lin. Augmenting panoramas with object movies by generating novel views with disparity-based view morphing. *Journal of Visualization and Computer Animation*, 2002.

[Hir06] H. Hirschmüller. Stereo vision in structured environments by consistent semi-global matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[Hir07] H. Hirschmüller. Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[HM01] Z. He and S. Mitra. A unified rate-distortion analysis framework for transform coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2001.

[Hop96] H. Hoppe. Progressive meshes. *SIGGRAPH '96*, 1996.

[Hor87] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 1987.

[HPDvG99] B. Heigl, M. Pollefeys, J. Denzler, and L. van Gool. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. *Symposium für Mustererkennung*, 1999.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. *Journal of the Optical Society of America*, 1988.

[HS04] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding*, 2004.

[Huf52] D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proceeding Institute of Electrical and Radio Engineers*, 1952.

[HZ04] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2004.

[IJ92] ITU-T and ISO/IEC JTC1. Digital compression and coding of continuoustone still images. *ISO/IEC 10918-1 - ITU-T Recommendation T.81 (JPEG)*, 1992.

[IJ94] ITU-T and ISO/IEC JTC 1. Generic coding of moving pictures and associated audio information - part 2: Video. *ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2)*, 1994.

[IMG99] A. Isaksen, L. McMillan, and L. Gortler. Dynamically reparameterized light fields. *Technical Report MIT-LCS-TR-778*, 1999.

[IPL97] I. Ihm, S. Park, and R.K. Lee. Rendering of spherical light fields. *Conference on Computer Graphics and Applications*, 1997.

[IS03] J. Isidoro and S. Sclaroff. Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints. *IEEE International Conference on Computer Vision*, 2003.

[ITJ94] ITU-T and ISO/IEC JTC 1. Generic coding of moving pictures and associated audio information - part 2: Video. *ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2)*, 1994.

[ITU00] ITU-T. Video coding for low bit rate communication. *ITU-T Rec. H.263; v1: Nov. 1995, v2: Jan. 1998, v3: Nov. 2000*, 2000.

[IW05] A. Ilie and G. Welch. Ensuring color consistency across multiple cameras. *IEEE Conference on Computer Vision*, 2005.

[Joi03] Joint Video Team of ITU-T and ISO/IEC JTC 1. Draft ITU-t recommendation and final draft international standard of joint video specification (ITU-t rec. H.264 ISO/IEC 14496-10 AVC). *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050*, 2003.

[JSA03] A. Jagmohan, A. Seghal, and N. Ahuja. Compression of light-field rendering data using coset codes. *ASILOMAR Conference on Signals and Systems*, 2003.

[JVT03] JVT Joint Video Team of ITU-T and ISO/IEC JTC 1. Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.264 ISO/IEC 14496-10 AVC). *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050*, 2003.

[JWV+05] N. Joshi, B. Wilburn, V. Vaish, M. Levoy, and M. Horowitz. Automatic color calibration for large camera arrays. *UCSD CSE Tech Report CS2005-0821*, 2005.

[KCTS01] R. Krishnamurthy, B.B. Chai, H. Tao, and S. Sethuraman. Compression and transmission of depth maps for image-based rendering. *International Conference on Image Processing*, 2001.

[KIS01] H. Kawasaki, K. Ikeuchi, and M. Sakauchi. Light field rendering for large-scale scenes. *International Conference on Computer Vision and Pattern Recognition*, 2001.

[KK03] M. Karczewicz and R. Kurceren. The SP- and SI-frames design for H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[KPG00] R. Koch, M. Pollefeys, and L. J. Van Gool. Realistic surface reconstruction of 3D scenes from uncalibrated image sequences. *Journal of Visualization and Computer Animation*, 2000.

[KS04] Sing Bing Kang and Richard Szeliski. Extracting view-dependent depth maps from a collection of images. *International Journal on Computer Vision*, 2004.

[KTAC04] A. Kubota, K. Takahashi, K. Aizawa, and T. Chen. All-focused light field rendering. *Eurographics Symposium on Rendering*, 2004.

[KWLS03] S.B. Kang, M. Wu, Y. Li, and H.-Y. Shum. Large environment rendering using plenoptic primitives. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[KZ02] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *European Conference on Computer Vision*, 2002.

[LA04] M.I.A. Lourakis and Antonis A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. *Technical Report FORTH-ICS*, 2004.

[Lau94] A. Laurentini. The visual hull concept for silhouette based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994.

[LCM+06] Yang Liu, G. Chen, N. Max, C. Hofsetz, and P. McGuiness. Undersampled light field rendering by a plane sweep. *Computer Graphics Forum*, 2006.

[Len98] J. Lengyel. The convergence of graphics and vision. *IEEE Computer Society Press*, 1998.

[Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 1944.

[LF94] S. Laveau and O. Faugeras. 3D scene representation as a collection of images. *International Conference on Pattern Recognition*, 1994.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981.

[LH96] M. Levoy and P. Hanrahan. Light field rendering. *SIGGRAPH '96*, 1996.

[Lip80] A. Lippman. Movie maps: An application of the optical videodisk to computer graphics. *SIGGRAPH '80*, 1980.

[LIZ+04] M. L. Levkovich, A. Ignatenko, A. Zhirkov, A. Konushin I.K. Park, M. Han, and Y. Bayakovski. Depth image-based representation and compression for static and animated 3D objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004.

[LNA+06] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. *SIGGRAPH '06*, 2006.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[LPC+00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, L. Pereira D. Koller, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. *SIGGRAPH '00*, 2000.

[LS00] Z.-C. Lin and H.-Y. Shum. On the numbers of samples needed in light field rendering with constant-depth assumption. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000.

[LSG02] R. Luetolf, B. Schiele, and M. H. Gross. The light field oracle. *Pacific Conference on Computer Graphics and Applications*, 2002.

[LWLZ02b] L. Luo, Y. Wu, J. Li, and Y.-Q. Zhang. 3D wavelet compression and progressive inverse wavelet synthesis rendering of concentric mosaic. *IEEE Transactions on Image Processing*, 2002.

[LZWS00] J. Li, K. Zhou, Y. Wang, and H.-Y. Shum. A novel image-based rendering a longitudinally aligned camera array. *Eurographics Workshop on Rendering*, 2000.

[Mag05] M. Magnor. Video-based rendering. *A K Peters*, 2005.

[Mar63] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Society for Industrial and Applied Mathematics*, 1963.

[MBR+00] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. *SIGGRAH '00*, 2000.

[McM99] L. McMillan. An image-based approach to three-dimensional computer graphics. *PhD thesis, University of North Carolina, Chapel Hill*, 1999.

[MEG00a] M. Magnor, P. Eisert, and B. Girod. Model-aided coding of multi-viewpoint image data. *IEEE International Conference on Image Processing*, 2000.

[MG95] L. McMillan and G.Bishop. Plenoptic modeling: An image-based rendering system. *SIGGRAPH '95*, 1995.

[MG99] M. Magnor and B. Girod. Fully embedded coding of triangle meshes. *Vision, Modeling, and Visualization*, 1999.

[MG00a] M. Magnor and B. Girod. Data compression for light-field rendering. *IEEE Transactions on Circuits and Systems for Video Technology*, 2000.

[MM05] B. Mercier and D. Meneveaux. Shape from silhouette: Image pixels for marching cubes. *International Conferences on Computer Graphics, Visualization and Computer Vision*, 2005.

[MPZ+02] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan. Acquisition and rendering of transparent and refractive objects. *Eurographics Workshop on Rendering*, 2002.

[MRG03] M. Magnor, P. Ramanathan, and B. Girod. Multi-view coding for image-based rendering using 3D scene geometry. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[MRP98] G. Miller, S. Rubin, and D. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. *Eurographics Rendering Workshop*, 1998.

[MT96] R. Mohr and B. Triggs. Tutorial on projective geometry. *International Symposium of Photogrammetry and Remote Sensing*, 1996.

[NLB+05] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Technical Report CSTR 2005-02, Computer Science Department, Stanford University*, 2005.

[Nyq28] H. Nyquist. Certain topics in telegraph transmission theory. *Trans. AIEE*, 1928.

[OK85] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1985.

[PAHL90] A. Puri, R. Aravind, B.G. Haskell, and R. Leonardi. Video coding with motion-

compensated interpolation for cd-rom applications. *Signal Processing: Image Communication*, 1990.

[PBE99] S. Peleg and M. Ben-Ezra. Stereo panorama with a single camera. *Computer Vision and Pattern Recognition Conference*, 1999.

[PCD+97] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. *Eurographics Workshop on Rendering*, 1997.

[Pol00] M. Pollefeys. Obtaining 3D models with a hand-held camera/3D modeling from images. *European Conference on Computer Vision*, 2000.

[PS01] I. Peter and W. Strasser. The wavelet stream: Interactive multi resolution light field rendering. *Eurographics Workshop on Rendering*, 2001.

[PSM03] R. Pajarola, M. Sainz, and Y. Meng. Depth-mesh objects: Fast depth-image meshing and warping. *UCI-ICS Technical Report No. 03-02, University of California, Irvine*, 2003.

[Rad99] P. Rademacher. View-dependent geometry. *SIGGRAPH '99*, 1999.

[RFG01] P. Ramanathan, M. Flierl, and B. Girod. Multi-hypothesis prediction for disparity compensated light field compression. *IEEE International Conference on Image Processing*, 2001.

[RG02] P. Ramanathan and B. Girod. Theoretical analysis of geometry inaccuracy for light field compression. *IEEE International Conference on Image Processing*, 2002.

[RG04a] P. Ramanathan and B. Girod. Random access for compressed light fields using multiple representations. *IEEE International Workshop on Multimedia Signal Processing*, 2004.

[RG04b] P. Ramanathan and B. Girod. Rate distortion optimized streaming of compressed light fields with multiple representations. *Packet Video Workshop*, 2004.

[RG05] P. Ramanathan and B. Girod. Receiver-driven rate-distortion optimized streaming of light fields. *IEEE International Conference on Image Processing*, 2005.

[RKG03] P. Ramanathan, M. Kalman, and B. Girod. Rate distortion optimized streaming of compressed light fields. *IEEE International Conference on Image Processing*, 2003.

[RKG07] P. Ramanathan, M. Kalman, and B. Girod. Rate-distortion optimized interactive light field streaming. *IEEE Transactions on Multimedia*, 2007.

[SCD+06] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *Computer Vision and Pattern Recognition*, 2006.

[SCG97] P.-P. J. Sloan, M. F. Cohen, and S. J. Gortler. Time critical lumigraph rendering. *Symposium on Interactive 3D Graphics*, 1997.

[Sch07] F. Schweiger. Calibration of a heterogeneous multi-sensor platform. *Diploma Thesis, TU München, Fachgebiet Medientechnik*, 2007.

[SCK07] H.-Y. Shum, S.-C. Chan, and S. B. Kang. *Image-Based Rendering*. 2007.

[SCMS04] G. Slabaugh, B. Culbertson, T. Malzbender, and M. Stevens. Methods for volumetric reconstruction of visual scenes. *International Journal of Computer Vision*, 2004.

[SD96] S.M. Seitz and C. M. Dyer. View morphing. *SIGGRAPH '96*, 1996.

[SD97] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Conference on Computer Vision and Pattern Recognition*, 1997.

[SFLG00] K. Stuhlmüller, N. Färber, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communications*, 2000.

[SGHS98] J. W. Shade, S. J. Gortler, L.-W. He, and R. Szeliski. Layered depth images. *SIGGRAPH '98*, 1998.

[SH99] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. *SIGGRAPH '99*, 1999.

[Sha48] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 1948.

[Sha59] C.E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, 1959.

[SHS99] H. Schirmacher, W. Heidrich, and H.-P. Seidel. Adaptive acquisition of lumigraphs from synthetic scenes. *EUROGRAPHICS '99*, 1999.

[SKC03] H.-Y. Shum, S.-B. Kang, and S.-C. Chan. Survey of image-based representations and compression techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[SL05] A.M.K. Siu and R.W.H. Lau. Image registration for image-based rendering. *IEEE Transactions on Image Processing*, 2005.

[SLKS05] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[SM99] P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[SMP05] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 2005.

[SNC05] H.-Y. Shum, K.-T. Ng, and S.-C. Chan. A virtual reality system using the concentric mosaic: Construction, rendering, and data compression. *IEEE Transactions on Multimedia*, 2005.

[SS90] V. Salari and I. K. Sethi. Feature point correspondence in the presence of occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 2002.

[SSS06] N. Snavely, S. Seitz, , and R. Szeliski. Photo tourism: exploring photo collections in 3D. *SIGGRAPH '06*, 2006.

[SSY+04] H.-Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C.-K. Tang. Pop-up light field: An interactive image-based modeling and rendering system. *ACM Transactions on Graphics*, 2004.

[ST96] P. Sturm and B. Triggs. Factorization based algorithm for multi-image projective structure and motion. *European Conference on Computer Vision*, 1996.

[ST01] A. Secker and D. Taubman. Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting. *IEEE International Conference on Image Processing*, 2001.

[Stu97] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[SW98] G. J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 1998.

[SWG99] E. Steinbach, T. Wiegand, and B. Girod. Using multiple global motion models for improved block-based video coding. *IEEE International Conference on Image Processing*, 1999.

[SYGM03] J. Stewart, J. Yu, S. J. Gortler, and L. McMillan. A new reconstruction filter for undersampled light fields. *Eurographics Workshop on Rendering*, 2003.

[SZS03] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[TAL+07] C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, and H.-P. Seidel. Seeing people in different light - joint shape, motion and refectance capture. *IEEE Transactions on Visualization and Computer Graphics*, 2007.

[TG00] X. Tong and R.M. Gray. Coding of multi-view images for immersive viewing. *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2000.

[TG03] X. Tong and R.M. Gray. Interactive rendering from compressed light fields. *IEEE Trans. on Circuits and Systems for Video Technology*, 2003.

[TK95] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.

[TKN03] K. Takahashi, A. Kubota, and T. Naemura. All in-focus view synthesis from under-sampled light fields. *International Conference on Artificial Reality and Telexistence*, 2003.

[TM02] D. S. Taubman and M. W. Marcellin. JPEG2000: Image compression fundamentals, standards, and practice. *Kluwer Academic Publishers*, 2002.

[TMHF99] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment a modern synthesis. *Vision Algorithms: Theory and Practice*, 1999.

[TN06] K. Takahashi and T. Naemura. A theory of aliasing separation for light field data. *IEEE International Conference on Image Processing*, 2006.

[Tri98] B. Triggs. Autocalibration from planar scenes. *European Conference on Computer Vision*, 1998.

[TSK+01] X. Tong, H.-Y. Shum, S.B. Kang, T. Feng, and R. Szeliski. Locally reparameterized light fields. *SIGGRAPH '01*, 2001.

[TTV+02] T.Werner, T.Pajdla, V.Hlavac, A.Leonardis, and M.Matousek. Selection of reference images for image-based representation. *Computing*, 2002.

[UH05] R. Unnikrishnan and M. Hebert. Fast extrinsic calibration of a laser rangefinder to a camera. *Technical Report CMU-RI-TR-05-09, Carnegie Mellon University*, 2005.

[UT03] T. Ueshiba and F. Tomita. Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices. *IEEE International Conference on Computer Vision*, 2003.

[VBK05] S. Vedula, S. Baker, , and T. Kanade. Image based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 2005.

[vdSA05] M. van der Schaar and Y. Andreopoulos. Rate-distortion-complexity modeling for network and receiver aware adaptation. *IEEE Transactions on Multimedia*, 2005.

[VFSH04] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum*, 2004.

[VNP02] J. Valentim, P. Nunes, and F. Pereira. Evaluating MPEG-4 video decoding complexity for an alternative video complexity verifier model. *IEEE Transactions On Circuits And Systems For Video Technology*, 2002.

[VWJL04] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy. Using plane + parallax for calibrating dense camera arrays. *International Conference on Computer Vision and Pattern Recognition*, 2004.

[WAA+00] D.N. Wood, D.I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D.H. Salesin, and W. Stuetzle. Surface light fields for 3D photography. *SIGGRAPH 2000*, 2000.

[Wan95] B. Wandell. Foundations of vision. *Sinauer Associates, Inc.*, 1995.

[WCH92] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[WJV+05] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 2005.

[WNC87] I.H. Witten, R.M. Neal, and J.G. Cleary. Arithmetic coding for data compression. *ACM Communication Association for Computing Machinery*, 1987.

[WOZ02] Y. Wang, J Ostermann, and Y.-Q. Zhang. Video processing and communications. *Prentice-Hall*, 2002.

[WSLH02] B. Wilburn, M. Smulski, H.-H. K. Lee, and M. Horowitz. The light field video camera. *SPIE Electronic Imaging*, 2002.

[Wys58] G. Wyszecki. Evaluation of metameric colors. *Journal of the Optical Society of America (1917-1983)*, 1958.

[YEBM02] J. C. Yang, M. Everett, C. Buehler, and L. McMillan. A real-time distributed light field camera. *Eurographics Workshop on Rendering*, 2002.

[ZAG03] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. *IEEE Workshop on Statistical Signal Processing*, 2003.

[ZBTC06] P. Zanuttigh, N. Brusco, D. Taubman, and G. Cortelazzo. A novel framework for the interactive transmission of 3D scenes. *Special Issue on Interactive Representation Of Still and Dynamic Scenes*, 2006.

[ZC03a] C. Zhang and T. Chen. Non-uniform sampling of image-based rendering data

with the position-interval error (PIE) function. *International Conference on Visual Communications and Image Processing*, 2003.

[ZC03b] C. Zhang and T. Chen. Spectral analysis for sampling image-based rendering data. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[ZC04] C. Zhang and T. Chen. A survey on image-based rendering - representation, sampling and compression. *EURASIP Signal Processing: Image Communication*, 2004.

[ZC07] C. Zhang and T. Chen. Active rearranged capturing of image-based rendering scenes - theory and practice. *IEEE Transactions on Multimedia*, 2007.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[Zha04] C. Zhang. On sampling of image-based rendering data. *PhD Thesis, Carnegie Mellon University*, 2004.

[Zhi01] A. Zhirkov. Binary volumetric octree representation for image based rendering. *GRAPHICON*, 2001.

[ZK07] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 2007.

[ZKU+04] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 2004.

[ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 1977.

[ZL00] C. Zhang and J. Li. Compression of lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering. *Proceedings of IEEE Data Compression Conference*, 2000.

[ZL01] C. Zhang and J. Li. Interactive browsing of 3D environment over the internet. *SPIE Visual Communication and Image Processing*, 2001.

[ZL05] C. Zhang and J. Li. On the compression and streaming of concentric mosaic data for free wandering in a realistic environment over the internet. *IEEE Transactions on Multimedia*, 2005.

[ZP04a] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). *International Conference on Intelligent Robots and Systems*, 2004.

[ZP04b] Q. Zhang and R. Pless. Fusing video and sparse depth data in structure from motion. *International Conference on Image Processing*, 2004.

Cited websites

[Bou07] J.-Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calibdoc/, last modified July, 2007.

[Gar06] M. Garland. Qslim. http://graphics.cs.uiuc.edu/garland/software/qslim10.html, last modified November, 2006.

[IDS07] IDS Imaging Development Systems GmbH. Product Page. http://www.ids-imaging.de/, last modified August, 2007.

[RIE07] RIEGL. LMS-Z420i. http://www.riegl.com/, last modified July, 2007.

[SFA07] W. Sepp, S. Fuchs., and K. Arbter. DLR CalLab and CalDe - the DLR camera calibration toolbox, last modified july, 2007. http://www.dlr.de/rm-neu/desktopdefault.aspx/tabid-3925/, last modified July, 2007.

[SIC07] SICK AG. SICK LMS 291S5. http://www.sick.com/home/en.html, last modified July, 2007.

[SS07] D. Scharstein and R. Szeliski. Middlebury Stereo Vision Home Page. http://vision.middlebury.edu/stereo/data/, last modified July, 2007.