Diss. ETH No. 22933

3D Reconstruction and Rendering from High Resolution Light Fields

A thesis submitted to attain the degree of
Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by
Changil Kim
MSc in Computer Science, ETH Zurich, Switzerland
born on June 19, 1979
citizen of the Republic of Korea

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Dr. Alexander Sorkine-Hornung, co-examiner
Prof. Dr. Brian Curless, co-examiner

2015

DISS. ETH NO. 22933

3D RECONSTRUCTION AND RENDERING FROM HIGH RESOLUTION LIGHT FIELDS

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

CHANGIL KIM

MSc in Computer Science, ETH Zurich

born on
June 19, 1979

citizen of
The Republic of Korea

accepted on the recommendation of

Prof. Dr. Markus Gross
Dr. Alexander Sorkine-Hornung
Prof. Dr. Brian Curless

2015

Abstract

This thesis presents a complete processing pipeline of densely sampled, high resolution light fields, from acquisition to rendering. The key components of the pipeline include 3D scene reconstruction, geometry-driven sampling analysis, and controllable multiscopic 3D rendering.

The thesis first addresses 3D geometry reconstruction from light fields. We show that dense sampling of a scene attained in light fields allows for more robust and accurate depth estimation without resorting to patch matching and costly global optimization processes. Our algorithm estimates the depth for each and every light ray in the light field with great accuracy, and its pixel-wise depth computation results in particularly favorable quality around depth discontinuities. In fact, most operations are kept localized over small portions of the light field, which by itself is crucial to scalability for higher resolution input and also well suited for efficient parallelized implementations. Resulting reconstructions retain fine details of the scene and exhibit precise localization of object boundaries.

While it is the key to the success of our reconstruction algorithm, the dense sampling of light fields entails difficulties when it comes to the acquisition and processing of light fields. This raises a question of optimal sampling density required for faithful geometry reconstruction. Existing works focus more on the alias-free rendering of light fields, and geometry-driven analysis has seen much less research effort. We propose an analysis model for determining sampling locations that are optimal in the sense of high quality geometry reconstruction. This is achieved by analyzing the visibility of scene points and the resolvability of depth and estimating the distribution of reliable estimates over potential sampling locations.

A light field with accurate depth information enables an entirely new approach to flexible and controllable 3D rendering. We develop a novel algorithm for multiscopic rendering of light fields which provides great controllability over the perceived depth conveyed in the output. The algorithm synthesizes a pair of stereoscopic images directly from light fields and allows us to control stereoscopic and artistic constraints on a per-pixel basis. It computes non-planar 2D cuts over a light field volume that best meet described constraints by minimizing an energy functional. The output images are synthesized by sampling light rays on the cut surfaces. The algorithm generalizes for multiscopic 3D displays by computing multiple cuts.

The resulting algorithms are highly relevant to many application scenarios. It can readily be applied to 3D scene reconstruction and object scanning, depth-assisted segmentation, image-based rendering, and stereoscopic content creation and post-processing, and can also be used to improve the quality of light field rendering that requires depth information such as super-resolution and extended depth of field.

Zusammenfassung

Diese Dissertation präsentiert eine komplette Verarbeitungspipeline für dicht abgetastete, hochauflösende Lichtfelder, von der Akquisition bis zu deren Rendering. Die wichtigsten Komponenten dieser Pipeline umfassen 3D-Szenenrekonstruktion, Geometrie-gesteuerte Abtastanalyse, und kontrollierbares multiskopisches 3D-Rendering.

Diese Dissertation beschäftigt sich zunächst mit 3D-Geometrierekonstruktion von Lichtfeldern. Wir zeigen, dass dichte Abtastung einer Szene in Form von Lichtfeldern eine robuste und genaue Tiefenmessung ermöglicht, und zwar ohne Nachbarschaften von Pixeln zu vergleichen, und ohne auf kostspielige globale Optimierungsprozesse zurückzugreifen. Unser Algorithmus schätzt die Tiefe für jeden Lichtstrahl im Lichtfeld mit grosser Genauigkeit, und die resultierenden pixelweisen Tiefenwerte sind von besonders hoher Qualität in der Nähe von Tiefendiskontinuitäten. In der Tat sind die meisten Operationen für kleine Abschnitte des Lichtfeldes lokalisiert, was entscheidend für die Skalierbarkeit für höher aufgelöste Eingabedaten ist und auch eine effiziente parallele Implementierung erlaubt. Die resultierenden Rekonstruktionen erhalten feine Seznendetails mit einer präzisen Lokalisierung von Objektkanten.

Während es der Schlüssel zum Erfolg unseres Rekonstruktionsalgorithmus ist, führt die hohe Abtastdichte von Lichtfeldern zu Problemen bei deren Aufnahme und Verarbeitung. Dies wirft die Frage der optimalen Abtastsdichte auf welche zur genauen Geometrierekonstruktion erforderlich ist. Bestehende Arbeiten konzentrieren sich mehr auf das Alias-freie Rendering von Lichtfeldern und Geometrie-getriebene Analyse hat deutlich weniger Forschungsanstrengungen

gesehen. Wir schlagen ein Analysemodell zur Bestimmung von Aufnahmepositionen vor, die im Sinne von qualitativ hochwertigen Geometrierekonstruktion optimal sind. Dies wird durch die Analyse der Sichtbarkeit der Szenenpunkte und der Tiefenauflösbarkeit, sowie der Bestimmung der Verteilung der zuverlässigen Schätzungen für potenzielle Aufnahmepositionen erreicht.

Ein Lichtfeld mit genauer Tiefeninformation ermöglicht einen völlig neuen Ansatz für flexibles und steuerbares 3D-Rendering. Wir entwickeln einen neuartigen Algorithmus für multiskopisches Rendering von Lichtfeldern, der grosse Steuerbarkeit über die wahrgenommene Tiefe in der Ausgabe erlaubt. Der Algorithmus synthetisiert ein Paar von stereoskopischen Bildern direkt von Lichtfeldern und ermöglicht es, stereoskopische und künstlerische Vorgaben auf Pixelebene zu steuern. Er berechnet nicht-planare 2D-Schnitte in einem Lichtfeldvolumen, welche die vorgegebenen Vorgaben bestmöglich erfüllen durch die Minimierung eines Energiefunktionals. Die Ausgangsbilder werden durch Abtasten von Lichtstrahlen auf den Schnittflächen synthetisiert. Der Algorithmus kann für multiskopische 3D-Displays verallgemeinert werden indem mehrere Schnitte berechnet werden.

Die resultierenden Algorithmen sind von grosser Bedeutung für viele Anwendungsszenarien. Sie sind leicht anwendbar für 3D-Szenenrekonstruktion und Objektscanning, Tiefen-gestützte Segmentierung, bildbasiertes Rendering sowie stereoskopische Inhaltsgenerierung und Nachbearbeitung, und können auch verwendet werden, um die Qualität von Lichtfeld-Rendering zu verbessern, welches Tiefeninformation erfordert, wie zum Beispiel Super-Resolution und erweiterte Tiefenschärfe.

Acknowledgments

I would like to thank my advisor and mentor Markus Gross. He was inspiring, enthusiastic about pushing the boundaries of computer graphics research, and allowed me excellent research environments both at ETH and Disney Research. Without him, this thesis would not exist. I am grateful to Alexander Sorkine-Hornung for sharing the burden of advising me. He supported my work, opened many exciting opportunities, and showed great insights and leadership throughout my PhD. I am thankful to Brian Curless for serving on my thesis committee and reviewing my thesis.

Nothing in this thesis would have been realized without my coauthors. Thank you very much, it was a great pleasure to work with you all: Simon Heinzle, Wojciech Matusik, Henning Zimmer, Yael Pritch, Ulrich Müller, Kartic Subr, and Kenny Mitchell. Maurizio Nitti and Alessia Marra have been patient whenever having to deal with my requests for new datasets that might have seemed nonsensical to them. I am also grateful to Manuel Lang, Thomas Oskam, Wojciech Jarosz, Thabo Beeler, Paul Beardsley and Team Skye, and Katie Basset for allowing me to use their datasets, or voice. Thank you, Henning, for proofreading my thesis.

I was lucky enough to be affiliated with two great institutions, ETH and Disney Research, and to have worked at both. I would like to thank all members of the Computer Graphics Laboratory and the Interactive Geometry Lab at ETH, and Disney Research. I have learned and benefited so much from them both professionally and personally.

Research would have been less exciting without bzflag—our little, secretive team building activity: Alex, Henning, Oliver Wang, Jean-Charles Bazin, Kaan Yücer,

and Federico Perazzi. Oh and Ulrich. Of course, they taught me so much more than bzflag.

Thank you all who showed up in the events I shamelessly invited to. In particular, those who frequented: Alec Jacobson, Mélina Skouras, Nicolas Chignardet, Antoine Milliez, Kenshi Takayama, and Jean-Charles.

All the further years at Disney Research started from the internship there. Thank you very much, Steven Poulakos and Jeroen van Baar, for giving me this opportunity. Thank you, Simon and Wojciech Matusik, for leading me to the topics of light fields and eventually fruitful outcomes. It was a great pleasure to supervise and work with talented students and interns: Christian Reiter, Ulrich Müller, Werner Randelshofer, Guo Qi, and Matan Zohar. I am thankful to David DiFrancesco, Beth Sullivan, Tom Duff, and Mark VandeWettering for hosting me at Pixar. I feel indebted whenever I see hex keys. When I was stuck with peculiar behaviors of whatever software I used, Gerhard Röthlin was my messiah. Andreas Baumann saved my laptop, thereby my hours and days, several times. Tobias Nägeli generously fixed the linear stage we used to capture datasets. I am also thankful to our administrators: Lioudmila Thalmann, Sarah Disch, Markus Portmann, and Martina Haefeli. Denise Spicher has always been a great supporter of all students, and I was not an exception.

I would like to express my deepest gratitude to my mother, father, and sister for their unconditional love and support. They gave me the courage to change my career entirely, to move to a new world, and to stand there on my own. Mom, you always believed that one day I'd be a PhD. Thank you, Mom.

Since the very beginning of my PhD, my wife has always been with me, as the best friend when I had hard times or good times as well, as the one who cheered me up and made me always smile, and as the greatest supporter who enabled me to go through those intensive years. Woo, I am infinitely grateful to you.

Contents

Introduction **1**
1.1 Contributions 3
1.2 Organization 4
1.3 Publications 5

Related Work **7**
2.1 A Brief History of Light Fields 7
2.2 Light Field Acquisition 8
2.3 Geometry Reconstruction 10
2.4 Sampling Analysis 12
2.5 Light Field Rendering 14
2.6 Stereoscopic Rendering 15

Acquisition **19**
3.1 Light Field Parameterization 19
3.1.1 The Plenoptic Function 19
3.1.2 4D Light Fields 20
3.1.3 3D Light Fields and EPIs 21
3.1.4 Notational Conventions 22
3.2 Light Field Acquisition 23
3.2.1 Camera Arrays and Gantries 23
3.2.2 Light Field Cameras 24
3.2.3 Unstructured Light Fields 25
3.3 Capture and Calibration 27

3.3.1 Capture Using a Linear Stage . . . 27
3.3.2 Hand-Held Capture . . . 28
3.3.3 Post-processing of Captured Images . . . 28

Geometry Reconstruction **31**
4.1 Introduction . . . 32
4.2 Sparse Representation . . . 34
4.3 Depth Estimation . . . 36
4.3.1 Edge Confidence . . . 37
4.3.2 Depth Computation . . . 38
4.3.3 Depth Propagation . . . 41
4.3.4 Fine-to-Coarse Refinement . . . 41
4.3.5 Extension to 4D Light Fields . . . 43
4.3.6 Extension to Unstructured Light Fields . . . 43
4.4 Experimental Evaluation . . . 44
4.4.1 Results . . . 44
4.4.2 Comparisons . . . 46
4.4.3 Applications . . . 52
4.4.4 Results for 4D and Unstructured Light Fields . . . 53
4.5 Discussion . . . 56

Geometry-Driven Sampling Analysis **57**
5.1 Introduction . . . 57
5.2 Sampling Analysis Model . . . 59
5.2.1 Conservative Sampling Interval . . . 60
5.2.2 Determining Visible Intervals . . . 60
5.2.3 Depth Resolution of Correspondence Algorithms . . . 61
5.2.4 Combining Visibility and Depth Resolution . . . 61
5.3 Online View Sampling Algorithm . . . 62
5.3.1 Initial Step . . . 62
5.3.2 Iterations . . . 62
5.3.3 Termination . . . 63
5.3.4 Priority Queue for Sampling Locations . . . 63
5.4 Experimental Results . . . 64
5.5 Discussion . . . 68
5.5.1 Parameter Selection . . . 68
5.5.2 Limitations and Future Work . . . 69

Rendering **71**
6.1 Introduction . . . 72
6.2 Goal-Based Stereoscopic View Synthesis . . . 74
6.2.1 Image Synthesis from Light Fields . . . 75

6.2.2 Stereoscopy from Light Fields 76
6.2.3 Formulation as Energy Minimization 80
6.2.4 Optimization via Graph Cuts 83
6.2.5 Extensions 84
6.2.6 Results 87
6.3 Variational Formulation for View Synthesis 93
6.3.1 Variational Formulation 93
6.3.2 Optimization via Primal-Dual Iterations 96
6.3.3 Experimental Results 98
6.3.4 Relation to Depth Computation 104
6.4 Discussion 105

Conclusion **107**
7.1 Recapitulation 107
7.2 Limitations and Future Work 108
7.2.1 Geometry Reconstruction 108
7.2.2 Sampling Analysis 109
7.2.3 Rendering 110

References **111**

Appendix: Curriculum Vitae **125**

1

Introduction

Since its introduction to computer graphics and vision, the concept of light fields has been widely adopted for many areas. It is one of the central representations for image-based rendering and 3D display techniques attributed to being conceptually simple while comprehensive. It serves as an invaluable tool for computational photography, extending the capability of conventional cameras and creating new applications such as digital refocusing. Popularized through the bullet time effect shown in movies like "The Matrix," light-field–based techniques are regularly practiced in movie productions, but also used in other industrial areas like manufacturing inspection.

In particular, image-based rendering using light fields has been established as an alternative to the traditional rendering pipeline based on 3D geometry and ray tracing. With a scene captured as a light field, one essentially has so many light rays at hand that rendering the scene from different perspectives reduces to simply picking relevant rays and interpolating between them. This can be attractive in many application scenarios since it is generally considered a hard problem to digitize the scene into an accurate 3D model and to render the acquired model photorealistically.

In its most general sense, a light field, also known as the plenoptic function, represents the *flow* of light at all positions in 3D space towards all directions over time. In geometrical optics, the flow is carried by rays and measured as radiance; a light field is a function that relates a ray to the radiance it transports. The ray is parameterized by a position, its direction, and time, making the light field a multi-dimensional function. Thus every collection of one or more photographs

of a scene is a particular sub-sampling of the light field, and conversely, low dimensional slices of a light field can be interpreted as some form of images, albeit not all of them will look like conventional photographs. For instance, one could consider Google Street View as one large light field. Pointing to a spot and looking around from there, one fixes the position (and also the time to when the location was captured) and gets a panoramic image at that point in time and space. Such conceptual flexibility and comprehensiveness make the light field suited to image-based rendering particularly well.

Although any collection of photographs constitutes a light field, we are often interested in more structured sampling of the light field, such as those captured using camera arrays or light field cameras. One characteristic observed in such sampling is a high level of coherency, redundancy, and smooth variations in the radiance of light fields. One of the central claims of this thesis is that such properties provide us with a new perspective to approach several important problems in computer graphics and vision. We support this claim by demonstrating examples of 3D geometry recovery and controllable 3D rendering, which together make up a complete rendering pipeline.

We first show that the coherency of dense light fields has unique properties that are beneficial to 3D geometry reconstruction, and mitigate difficulties arising with sparser sampling common in multi-view stereo setups. Dense directional sampling spanning a near continuum of baseline provides rich information about the trajectories of scene points according to changes in viewing positions. This works favorably for correspondence matching, reducing the need for larger patches to be compared to guarantee the required level of robustness. Our method uses a pixel-wise depth computation, which performs particularly well around depth discontinuities, advantageous for precise localization of object boundaries.

In addition, today's advances on imaging hardware allow us to image the world at much higher resolution with greater detail, while still many reconstruction techniques are not scalable to process high resolution data or not designed to deal with them. We propose a new concept we call fine-to-coarse refinement for regularized output, getting rid of expensive, often non-scalable global optimization from our pipeline without compromising the quality of results. Putting them together, we present a high quality 3D reconstruction framework, arguing that not only can the geometry reconstructed from light fields be used for improving rendering, but also for high quality 3D reconstruction itself. We show other direct applications as well including depth-assisted segmentation and free viewpoint rendering.

The problem of finding the right sampling rate, e.g., the required number of images and optimal capture locations, is important to minimize both the artifacts due to insufficient sampling and the amount of effort spent for sampling. For these

reasons this problem has been extensively studied to provide a sound and valid theoretical model for sampling light fields. However, most sampling analysis so far is targeted at reconstructing light fields for alias-free rendering, and does not provide an appropriate theory for sampling light fields in the context of geometry reconstruction. We address this and provide a theoretical foundation on the sampling analysis and an adaptive algorithm to sample light fields accordingly.

Lastly, it is demonstrated that light fields, along with the acquired geometry, allow us to render the scene in 3D with added controllability and flexibility over the perceived depth. Creating satisfactory 3D content fulfilling complicated display constraints and artistic requirements is not an easy task. Further, the content often has to be edited stereoscopically for displays of varying 3D capabilities or to achieve modified depth perception. Light fields are naturally suited to such tasks since they contain a wealth of information about spatial and directional variations of the scene. We present a method to synthesize stereoscopic image pairs directly from light fields given stereoscopic constraints and desired depth, which can be modified or specified at the user's disposal. The method is further extended for multi-view displays. While existing methods take sparser input and rely on image warping and inpainting, our method calculates the exact light rays needed, samples them from the light field, and composites them into complete images. Additionally, we provide a formulation that jointly solves for both depth estimation and content generation, removing the necessity of separate depth computation and additional storage, and running more efficiently.

1.1 Contributions

This thesis presents an image-based rendering pipeline for densely sampled light fields, from acquisition to rendering. It makes a number of technical contributions, which have led to several scientific publications and are summarized below:

- A novel geometry reconstruction algorithm tailored to densely sampled light fields. The algorithm makes use of the coherency in dense input and simplifies the correspondence matching problem dramatically while achieving higher quality reconstruction than other algorithms designed for sparser input. In particular, dense angular sampling allows the algorithm to use pixel-wise operations, which favor localizing object boundaries and fit greatly to today's parallelism in computing. This has led to our publication in 2013; see Section 1.3 for the details of the publication.
- A theory on sampling of light fields specifically targeting depth reconstruction. While in general dense sampling proves to be beneficial, it becomes more and more cumbersome to capture larger amounts of input, and at certain point no

additional improvement can be achieved despite increasing amounts of data. A natural question may be "how dense" the sampling should be. An answer is sought for this question via an analysis model on sampling and an algorithm to capture light fields according to this model. This led to a publication in 2015.

- An algorithm that renders stereoscopic content directly from a light field. The algorithm utilizes the acquired depth information and allows for accurate and flexible control over the perceived depth and stereoscopic viewing parameters, where effectively a set of such parameters can be specified for each pixel individually. Additionally, we present the derivation of a more efficient formulation incorporating both depth computation and rendering at once. This has led to two publications in 2011 and 2014, respectively.

1.2 Organization

The thesis is organized as follows:

- Chapter 2 reviews the literature that is closely connected to our research, in the area of light field acquisition, sampling analysis, geometry retrieval, and rendering.
- Chapter 3 formally introduces the notion of light fields and their mathematical representations, and discusses how light fields are acquired in practice using commodity imaging hardware. The conventions and notations we assume for the rest of the thesis are set in this chapter.
- Our the geometry reconstruction pipeline is presented in Chapter 4. This chapter also provides extensive comparisons against the existing reconstruction techniques as well as a number of immediate applications.
- Chapter 5 analyzes the sampling properties of light fields in the context of geometry reconstruction. A sampling analysis model is presented for this, and a sampling algorithm based on this model is proposed.
- Chapter 6 presents a framework for (multi-view) stereoscopic 3D rendering. Two formulations are proposed to synthesize stereoscopic views directly from light fields, along with ample results and comparisons.
- Finally, Chapter 7 concludes the thesis by recapitulating the core contributions and opening a few avenues for future research.

1.3 Publications

This thesis is based on the following peer-reviewed publications:

- C. KIM, A. HORNUNG, S. HEINZLE, W. MATUSIK, and M. GROSS. Multi-Perspective Stereoscopy from Light Fields. In *Proceedings of ACM SIGGRAPH Asia (Hong Kong, China, December 12–15, 2011)*, *ACM Transactions on Graphics*, vol. 30, no. 6, pp. 190:1–190:10, 2011.
- C. KIM, H. ZIMMER, Y. PRITCH, A. SORKINE-HORNUNG, and M. GROSS. Scene Reconstruction from High Spatio-Angular Resolution Light Fields. In *Proceedings of ACM SIGGRAPH (Anaheim, USA, July 21–25, 2013)*, *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 73:1–73:12, 2013.
- C. KIM, U. MÜLLER, H. ZIMMER, Y. PRITCH, A. SORKINE-HORNUNG, and M. GROSS. Memory Efficient Stereoscopy from Light Fields. In *Proceedings of International Conference on 3D Vision (Tokyo, Japan, December 8–11, 2014)*, pp. 73–80, 2014.
- C. KIM, K. SUBR, K. MITCHELL, A. SORKINE-HORNUNG, and M. GROSS. Online View Sampling for Estimating Depth from Light Fields. In *Proceedings of IEEE International Conference on Image Processing (Québec City, Canada, September 27–30, 2015)*, 2015 (to appear).

Although not directly related, the following peer-reviewed paper was published during the time period of this thesis:

- S. WENNER, J.-C. BAZIN, A. SORKINE-HORNUNG, C. KIM, and M. GROSS. Scalable Music: Automatic Music Retargeting and Synthesis. In *Proceedings of Eurographics (Girona, Spain, May 6–10, 2013)*, *Computer Graphics Forum*, vol. 32, no. 2, pp. 345–354, 2013.

2

Related Work

The idea of light fields was invented as early as in the dawn of the 20th century, although it is relatively recent when it was adopted in computer graphics and vision. This chapter reviews the literature related to light fields. We limit ourselves to the work closely connected to our research, since both the volume and the variety of existing work are enormous.

2.1 A Brief History of Light Fields

In 1908, Lippmann published two articles about *photographie intégrale*, translated literally as integral photography, which describes an imaging apparatus using small lenses arranged on a 2D grid that are able to capture multiple images of a scene with viewpoint variations [Lippmann, 1908a; Lippmann, 1908b]. The captured scene is reproduced in 3D as the viewer sees the parallax while the viewpoint changes. Since its invention, there had been many improvements of its design through a series of patents by many inventors, but it was much later when it started drawing attentions from research communities.

Adelson and Bergen [1991] proposed what they called the *plenoptic function* to systematically categorize the visual elements (stimuli) in early vision, which in combination, form visual information in the world. The plenoptic function represents the spectral radiance distribution of rays, and is defined as a multidimensional function of a position, an angular direction at the position, a wavelength, and a point in time. They cataloged the kind of visual stimuli as a local variation in one or more dimensions of the plenoptic function. Adelson and Wang [1992]

presented the design of the plenoptic camera where the light rays gathered through the main lens are recorded separately using a lenticular array placed on the sensor plane. They implemented the design using a set of relay lenses, and the light field recorded with this camera was used to obtain the scene depth by analyzing the directional variation of the radiance captured in the image.

In 1996, Levoy and Hanrahan [1996] and Gortler et al. [1996] proposed image-based rendering algorithms that are based on the representations they called the light field and the Lumigraph, respectively. Except for a few differences on the acquisition setups and the use of geometry proxy, the representation itself was largely in common, which is the now well-known two-plane light field. Further research immediately followed and the light field became available to many other areas, among which are computational photography [Veeraraghavan et al., 2007; Levin et al., 2008b; Levin et al., 2009] and computational 3D displays [Wetzstein et al., 2011; Lanman et al., 2011; Wetzstein et al., 2012]. Isaksen et al. [2000] studied the reparameterization of light fields, which became the basis of one of the best known applications of the light field—post-capture digital refocusing. Ng et al. [2005] presented the prototype of the first hand-held light field camera based on the design similar to that of Adelson and Wang, and demonstrated a few photographic effects such as an extended depth of field. This work led to Lytro, the first consumer light field camera.

2.2 Light Field Acquisition

A light field can be captured in various ways. Many of them rely on a controlled acquisition setup. Levoy and Hanrahan [1996] used in their paper a robotic gantry to position a camera to different viewing locations in a plane to sample the regular 4D ray space. Later researchers started using a few dozens of synchronized video cameras so that they can incorporate the temporal dimension and capture dynamic scenes. Yang et al. [2002] and Matusik and Pfister [2004] each built a system including acquisition, transmission, and rendering of light fields, using an array of video cameras and 3D displays, which amounts to a complete 3D TV system. Wilburn et al. [2005] developed a large-scale, high-performance camera array system, including 128 video cameras spanning about 1 meter horizontally and vertically. Joshi et al [2006] used a one-dimensional camera array and a motorized linear stage for their real-time matting system, which is similar to our acquisition setup we describe in Chapter 3. The size of such camera arrays varies from the one as tiny as a thumbnail [Venkataraman et al., 2013] that was deployed to mobile phones, to light domes that can accommodate a full human body with greater directional coverage [Kanade et al., 1997; Beeler et al., 2011; Joo et al., 2014].

While these acquisition devices can be built without having to design custom optics, they are often bulky and not portable, and require a huge data bandwidth to be dealt with. To address such problems, researchers tried to use conventional digital cameras. Ng et al. [2005] prototyped a hand-held light field camera, where they placed a micro-lens array on the sensor plane of a medium-sized camera to separate the light rays gathered by the camera's main lens. They used the matching f-number between each micro-lens and the main objective lens to maximize the use of sensor pixels. They demonstrate interesting applications using their prototype camera such as viewpoint change, refocusing, and all-in-focus imaging. Georgiev et al. [2006] explored along a similar direction, but instead of placing a regular lens array on the sensor plane, they placed a hexagonal array of lenses with varying focal lengths in front of the camera's main lens. Veeraraghavan et al. [2007] and Liang et al. [2008] used coded aperture techniques to computationally demultiplex the light rays collected through the camera's main lens at the price of reduced optical performance. Wetzstein et al. [2013] further discussed multiplexing light fields onto a 2D image sensor and developed a theory for multiplexing and a computational reconstruction algorithm. While these methods are usually more portable and able to capture light fields single-shot, they have an inherent problem: they have to share a single 2D image sensor to record both angular and spatial samples, thereby forced to trade between the resolutions of them. Currently, the designs based on micro-lenses are most common in the light field cameras on the market.

While the aforementioned methods rely on controlled acquisition setups, some strove for unstructured capture. Gortler et al. [1996] captured a collection of unstructured images and used them to populate the 4D ray space. They used markers to estimate the camera pose for each image and addressed the issues to fill the regular 4D grid data structure using unstructured input. Davis et al. [2012] further pursue this direction and proposed an interactive system which guides the user to orient the camera to capture the under-sampled part of the light field. There are hybrids of the structured and unstructured approaches; Zhang and Chen [2004] and Nomura et al. [2007] proposed reconfigurable, non-rigid camera arrays that allow the user to reshape or bend the camera array to meet a particular need of the user.

Although many such acquisition methods are proposed and built for specific applications in mind, design questions arising in the course are often answered based on experiences or empirical estimates, e.g., the required number of cameras and their locations, the resolution of cameras, etc. We address this issue later in Chapter 5, where we propose an analysis model that can help answer such questions.

A significant challenge of acquisition is that the captured set of images is very data-

intensive and also redundant. Thus, the early papers already discussed compact representations and compression schemes. Levoy and Hanrahan [1996] propose several representations for 4D light fields and apply a lossy vector quantization followed by entropy coding, while Gortler et al. [1996] applied standard image compression like JPEG to some of the views, and also pointed out the importance of depth information for sparser representation. Criminisi et al. [2005] investigated the segmentation of epipolar-plane images (EPIs) in 3D light fields into tubes representing layers of different objects. See Section 3.1.3 for an introduction to EPIs and 3D light fields. Storing colors and depth for each tube then gives a more compact representation of the light field. They also propose a method for detecting and removing specular highlights, but no solution for compactly storing this view-dependent information. Surface light fields [Wood et al., 2000; Chen et al., 2002] are an attractive solution to capture view-dependent effects, but they require accurate 3D geometry obtained by active scanning techniques. One component of our contribution is a sparse light field representation presented in Chapter 4 that differs from those previous approaches, fully reproduces the input light field including view dependent surface reflectance, and tightly integrates with our algorithm for depth estimation. The need for compact representation and efficient acquisition further motivates our analysis on light field sampling in Chapter 5.

2.3 Geometry Reconstruction

3D geometry reconstruction has been studied for decades and there is a huge body of research work. We focus on the reconstruction methods applicable for light fields, and cover methods in a broader context only briefly. One of the first approaches to extract depth from a dense sequence of images is the seminal work of Bolles et al. [Bolles et al., 1987; Bolles and Baker, 1987]. To our knowledge their technique is the first attempt to utilize the specific linear structures emerging in a densely sampled 3D light field for depth computation. However, the employed basic line fitting is not robust enough for a dense reconstruction of real world scenarios with occlusions, varying illumination, etc. and the reconstructions shown are sparse and noisy. On the other hand, many other methods adopt techniques from classical stereo reconstruction, i.e., matching corresponding pixels in all images of the light field, essentially using robust patch-based block matching [Zhang and Chen, 2004; Vaish et al., 2006; Bishop et al., 2009; Georgiev and Lumsdaine, 2010]. Along similar lines, Fitzgibbon et al. [2005] and Basha et al. [2012] describe robust clustering techniques to identify matching pixels. Ziegler et al. [2007] propose to analyze the Fourier spectra of EPIs sheared according to a hypothesized depth. As we demonstrate in our comparisons, such approaches often do not scale well to

high resolution light fields in terms of reconstruction quality and computational efficiency. Alternatively, one can extract depth from a light field using depth-from-(de)focus techniques such as Pentland [1987] and Grossmann [1987] by refocusing the light field at various focal planes and estimating the distance of the best focus, or even combining both approaches based on defocus blur and correspondence matching [Tao et al., 2013]. However, those methods face challenges similar to standard stereo approaches such as inaccuracies at silhouettes, but also have limitations due to the aperture size [Schechner and Kiryati, 2000].

In order to achieve higher overall coherence, various methods estimate depth as the minimizer of a global energy formulation where smoothness assumptions can be enforced [Adelson and Wang, 1992; Stich et al., 2006; Liang et al., 2008; Bishop and Favaro, 2010]. Notably, the recent energy-based approach of Wanner and Goldluecke [2012a] gives high quality depth maps from 4D light fields. But as for any global optimization method this comes at a very high computational cost. For example, the authors of the latter work report 10 minutes per single view depth map at 1 megapixel resolution. The direct application of such approaches to higher resolutions seems impractical. A second difficulty with approaches based on global optimization is to tune the underlying smoothness assumptions to preserve precise depth discontinuities at object contours, which are of highest importance in practice [Sylwan, 2010]. Fine details are often lost due to the involved coarse-to-fine multi-scale algorithms needed to avoid local minima. Our approach is particularly suited for such applications as it reconstructs precise depth estimates at the single pixel level, without the need for explicit global regularization; see more details in Chapter 4.

To illustrate the novel challenges arising from high resolution, densely captured light fields, we compare our results to some of currently best performing two- and multi-view stereo algorithms. For a more complete overview please refer to the evaluations of Scharstein and Szeliski [2002] and Seitz et al. [2006]. Despite considerable progress in this area [Kolmogorov and Zabih, 2001; Hirschmüller, 2005; Rhemann et al., 2011], with only two input views available one has to rely on complicated (patch) matching and some form of global smoothness. To alleviate over-smoothing of discontinuities, one can operate on larger image segments, or superpixels [Zitnick et al., 2004; Zitnick and Kang, 2007], but this may lead to over-segmentation artifacts in the depth maps at textured image regions. Also, with only a few views available, explicit detection and handling of occlusions is often required [Humayun et al., 2011; Ayvaci et al., 2012], which further increases the computational load. Some methods [Goldlücke and Magnor, 2003; Bleyer et al., 2011] jointly estimate depth and segmentation, but these again rely on costly global optimization. An alternative is to match a few reliable pixels only [Čech and Šára, 2007], and to densify the result later by spreading the sparse estimates [Sun et al., 2011]. However, existing approaches for sparse sample propagation generally

require a global energy minimization [Geiger et al., 2010], are prone to artifacts as shown in Szeliski and Scharstein [2002], or produce visually pleasing, but often physically distorted results [Lang et al., 2012]. Multi-view stereo techniques consider a larger number of images, spanning from tens [Seitz and Dyer, 1999; Kang and Szeliski, 2004; Zitnick et al., 2004; Vu et al., 2009; Beeler et al., 2010; Furukawa and Ponce, 2010] to several thousands [Snavely et al., 2008; Furukawa et al., 2010] to compute a more complete scene representation rather than single depth maps. However, these methods often provide either accurate but still sparse, or dense but over-smooth geometry and often do not scale well to very high resolution images. The coverage of the reconstructed scene with our method is higher than that of two-view stereo techniques, but lower than full 3D models generated with multi-view stereo. However, in contrast to the previously discussed techniques our algorithm produces a dense scene reconstruction with precise contours that is readily available for various applications such as novel view synthesis, depth-based segmentation, and other image-based applications.

2.4 Sampling Analysis

In computer graphics, two broad classes of reconstruction problems have been addressed from input of a collection of images (i.e., light rays). The goal of the first class, including light field rendering, is to reconstruct the view (i.e., a set of rays) from an arbitrary location from the measured values scattered in the domain. The second class, which includes geometry reconstruction, strives to infer the geometric structure of the scene from the measured radiance along light rays. This second class overlaps with multi-view stereo algorithms in computer vision. We concern ourselves with a general method of selecting the appropriate set of views which will be supplied to improve the fidelity of the reconstruction.

With alias-free rendering as its goal a substantial body of literature studied sample optimization strategies for light fields. Isaksen et al. [2000] and Gortler et al. [1996] address how to resample rays from already captured light fields for high quality rendering by reparameterizing light fields and using rough geometry, respectively. Lin and Shum [2000] provide an analysis given constant depth assumption, while Chai et al. [2000] further discuss the optimal sampling rate, e.g., the minimal number of views, when provided with either accurate or approximate depth in addition to the constant depth. Similarly, Shum et al. [2007] discuss light field sampling for reconstruction under alternative levels of provided depth information. Further, Zhang and Chen [2006] explore sampling analysis including reconfiguration of camera positions for improved render quality, for their reconfigurable camera array [Zhang and Chen, 2004]. Durand et al. [2005] explore more general physical phenomena regarding light transport and analyze them using Fourier

theory. Egan et al. [2011] derives the frequency analysis of occlusion in 4D ray space and present a filter to lower sampling rates for soft shadows. The focus of a target application area yields further opportunities for research, for example: Zwicker et al. [2006] analyze the light field signal towards an optimal reconstruction filter in the context of multi-view stereoscopic displays; Levin et al. [2008a; 2009; 2010] discuss the light field sampling for a wide variety of acquisition setups; and Davis et al. [2012] present an interactive light field acquisition system, where they determine the optimal sampling based on reprojection errors derived geometrically.

While it has been thoroughly studied for optimal rendering, the sampling property for optimal geometry recovery has not received as much attention. On the other hand, several previous works in robotics [Krainin et al., 2011], laser scanning [Maver and Bajcsy, 1993; Scott et al., 2003], shape recovery [Kutulakos and Dyer, 1994], and photogrammetry [Olague and Mohr, 2002] have pointed out the benefits of planning or selecting a *next best view* for improved localization, inspection, and reconstruction quality. Selecting an optimal camera separation is important for the quality of triangulation-based reconstruction methods. Okutomi and Kanade [1993] and Gallup et al. [2008] proposed to use variable baseline lengths depending on the hypothesized depth of a scene point, to improve the accuracy of triangulations. Vazquez et al. [2003] propose a framework similar to ours, where they develop a score that measures the amount of information seen from a viewpoint and determine the minimal number of viewpoints in a greedy manner. However, their score is motivated by image-based rendering while ours is based on geometry recovery. Also closely related are the *view selection* and *view clustering* in multi-view stereo, where one has to deal with a large collection of unstructured images and thus it is an important problem to maintain the computation tractable while not compromising the quality of reconstructions. For this, often employed strategies are to subsample images from a larger image collection [Goesele et al., 2007; Hornung et al., 2008], or cluster the images into several sets that can be processed in parallel [Furukawa et al., 2010], while minimizing the negative influence on the output quality. While these methods all achieve considerably improved results by targeting their strategies to the specific underlying algorithms, they often do not generalize well to other methods and do not always provide an extendable theoretic framework.

We provide a sampling analysis model motivated by geometry reconstruction from light fields, and propose a view sampling algorithm based on this model. Our analysis is based on trading off between two conflicting criteria of the visibility and the depth resolvability to determine whether the depth of a pixel can be estimated with enough accuracy. Our algorithm analyzes the very scene that is being captured, and estimates the distribution of pixels that can be faithfully reconstructed, to locate the best sampling positions.

2.5 Light Field Rendering

Since the light field was initially adopted for image-based rendering, there are many related research works. As discussed before, the two papers of Levoy and Hanrahan [1996] and Gortler et al. [1996] addressed many issues regarding light field rendering, including rendering algorithms, pre-filtering for anti-aliasing, and re-sampling and interpolation for novel view synthesis. In particular, Gortler et al. addressed the use of depth for higher quality rendering. Later, Isaksen et al. [2000] introduced synthetic aperture refocusing on fronto-parallel planes via reparameterization of light fields. Vaish et al. [2005] extended the refocusing to slanted planes. Shum and Kang [2000] review a broad range of early image-based rendering techniques according to how much geometric information is required. While previous techniques rely on regularly sampled light fields, i.e., those captured from cameras on a regular and planar grid, Buehler et al. [2001] presented a rendering algorithm that takes unstructured light fields as input. In particular, they integrated into a single framework two conceptually distant rendering algorithms, namely, the rendering of regularly sampled light fields with few geometric assumptions and the view-dependent texture mapping [Debevec et al., 1996] which requires relatively accurate geometric models but a smaller number of images. Davis et al. [2012] further extended it to an interactive system that acquires and renders unstructured light fields.

Availability of rough scene geometry can be used to achieve more faithful rendering. Isaksen et al. [2000] described how an approximate depth proxy may compensate sparse angular sampling, extending the initial idea of Gortler et al. [1996]. Similarly, Wanner et al. [2011] used a rough depth map to render light fields from a micro-lens array camera. Bishop et al. [2009] used depth information to super-resolve light fields. A few other methods reconstructed more elaborated 3D geometry proxies for rendering from a wider range of viewing positions. Zitnick et al. [2004] computed per-view depth maps for a layered scene representation and used border matting when warping each layer and compositing one over another. Hornung and Kobbelt [2009] proposed a GPU-accelerated particle rendering pipeline which uses per-view dense geometry proxies consisting of silhouette-aware particles. The rendering system integrates the particle cloud to generate output views at an interactive rate from novel viewing positions. An image-based rendering algorithm proposed by Chaurasia et al. [2013] uses superpixels, i.e., image over-segmentations, as its rendering primitives to deal with missing or unreliable depth information of the scene. These superpixels tend to honor object boundaries and hence depth discontinuities, and synthesized depth values are assigned to those in poorly reconstructed regions for plausible rendering with varying viewpoints.

In our rendering algorithm, novel images are synthesized directly from a light field.

These output images are created out of diverse rays contained in the light field, and effectively include the rays of multiple perspectives. This is the key to provide a high level of flexibility that allows for fulfilling complex output stereoscopic constraints.

In the history of art multi-perspective imaging has been used by painters and artists as a fundamental stylistic tool. Similar methods have later been employed by animators in movie production, e.g., for drawing backgrounds for 2D cell animation [Thomas and Johnston, 1995]. The computer graphics and computer vision community further studied the geometry and applications of multi-perspective imaging; a good overview is presented by Yu et al. [2010]. Wood et al. [1997] describe a first computer-assisted method to compute multi-perspective panoramas from a collection of perspective images. In the recent years many other types of multi-perspective cameras and corresponding images have been introduced: pushbroom cameras [Gupta and Hartley, 1997] and related multiple-center-of-projection images [Rademacher and Bishop, 1998], cross slit cameras [Pajdla, 2002; Zomet et al., 2003], or general linear cameras [Yu and McMillan, 2004]. In our work we do not assume any particular camera model. Instead the (multiple) perspectives of our images are optimized subject to prescribed stereoscopic disparity constraints.

The two most related publications to our algorithm are the works by Seitz [2001] and Peleg et al. [2001]. Seitz [2001] analyzes the space of all possible image types that provide depth cues due to binocular parallax, including multi-perspective images. He formally showed that epipolar geometry generalizes to multi-perspective images. His work provides a theoretical basis for our discussion of stereoscopic constraints and light field parameterization in Chapter 6. Peleg et al. [2001] provide a framework to construct multi-perspective omnidirectional stereoscopic images. Their method takes a video cube that captures a 360° panorama, and constructs two views that form a stereoscopic panorama with the disparity locally manipulated. They show how to dynamically adapt the baseline to modify scene parallax by a local selection scheme for image columns. Our work is inspired by these ideas and extends them to a more general and flexible framework using light fields, which generates globally optimal output views with respect to arbitrary, per-pixel disparity constraints.

2.6 Stereoscopic Rendering

This section briefly reviews existing techniques about stereoscopic rendering and content editing roughly in the order of increasing expressiveness. The readers interested in a broader range of related techniques are referred to recent surveys, such as Masia et al. [2013b].

The most basic means of disparity modification is to change the inter-axial distance, which is the distance between the two cameras' optical centers and also known as the baseline, and the convergence, the amount of rotation against each other around their vertical axes. The inter-axial distance scales the amount of perceived depth and the convergence translates the scene volume backward or forward when displayed, by shifting the plane of zero binocular parallax. Woods et al. [1993] identify such camera parameters that define the geometry of stereoscopic camera and display systems, and analyze the image distortions due to the variation of parameters. Jones et al. [2001] propose a method for controlling camera parameters by analyzing the scene depth range and mapping it to a given disparity budget. For this purpose, they separate the image space and the display space and provide a transformation between them. Holliman [2004] describes a system that compresses the scene depth for stereoscopic displays by identifying a region of interest and optimizing the perceived depth for this region compared to the rest of the scene. Zwicker et al. [2006] discuss how to optimize for the baseline and convergence by reparameterizing an input light field when rendering it for 3D automultiscopic displays. However, their concern is more on alias-free rendering rather then content editing, and their approach is to map the desired depth range to the in-focus range of the target display, blurring out the content outside the depth range.

In recent work, the control of baseline and convergence as well as other camera parameters, such as the field of view, camera movement, and so on, is almost fully automated according to the content of the scene about to be captured or rendered. The computational stereo camera of Heinzle et al. [2011] analyzes the scene it is capturing in real-time and adjusts those parameters, so that the captured scene remains in the stereoscopic comfort zone [Shibata et al., 2011]. Oskam et al. [2011] implement a similar idea in the context of real-time rendering such that the virtual scene is rendered in a fail-safe manner. Koppal et al. [2011] provide a detailed discussion on camera parameters for optimal stereo in their proposed shot-planning and post-production pipeline. Their tool converts the desired stereoscopic edits to optimal camera parameters. The control capability with only the camera parameters, however, is not enough for most application scenarios. Their expressive power is notably limited in that their change introduces a global effect over the entire screen space on the perceived depth, while in practice, more local control over stereoscopic depth perception is preferred. In addition, the parameters are determined with respect to a certain viewing condition, such as the screen size and viewing distance, and the content generated for one particular target cannot be easily adapted to different ones, requiring the same process to be redone for other viewing conditions.

Besides capturing the content stereoscopically, many techniques have been developed for creating stereoscopic content from existing 2D content. For computer-generated content, the scene depth is usually known and used to synthesize two

or more views for the target display. Given target disparities, the output images are rendered in several different manners, spanning from fast algorithms targeted for real-time applications to more sophisticated image warping techniques based on optimization and image decomposition. Bowles et al. [2012] proposed a fast image warping technique using fixed point iteration that is ideally targeted for real-time applications such as video games. Being part of the rendering pipeline, it has access to almost the complete information about the scene including the geometry, and the information additionally required can be rendered on demand. Along a similar line is the method of Didyk et al. [2010], which takes a single image and a depth buffer, and generates two views for the left and right eye using image-space adaptive grid warping at an interactive rate. Masia et al. [2013a] extend it to generate multiple output views to feed 3D automultiscopic displays. They also present a perceptually based disparity remapping that can compensate for the limited disparity bandwidth of 3D displays. Both methods use the same rendering technique, which handles disocclusions by stretching grid quads and may lead to visual artifacts.

While usually available in the animation pipeline for those method, precise depth information is not generally given for real-world content, and obtaining it from monoscopic input often requires manual interaction. Wang et al. [2011] propose an interactive user interface for the creation and manipulation of stereo content from existing 2D content, based on sparse user scribbles to annotate the scene depth that are propagated to fill the entire image space. However, in their method the resulting images are essentially warped versions of the original image, and thus often include noticeable distortions around the occlusion boundaries in particular. Ward et al. [2011] propose a system for 2D-to-3D conversion, where they use various computer vision and graphics techniques to aid the established workflow in movie productions based on rotoscoping and inpainting. However, the conversion essentially relies on image segmentation and warping, which are prone to errors, sharing the same problem described above, are not capable to handle view dependent effects such as specular highlights, and require intensive manual interaction.

For finer control over the stereo depth perception of existing 3D content, the usual strategy is to locally manipulate disparities of matching image features between two views. Several approaches have been proposed to implement such manipulation given particular requirements to the output stereoscopic images. The method of Lang et al. [2010] computes sparse correspondences between given two images and warps the images using a variational framework such that the correspondences will have modified parallax in the deformed image pair. To describe the desired artistic manipulation, they formally define a collection of disparity remapping operations, including nonlinear ones, which enable sophisticated control over disparity modification. Chang et al. [2011] proposed a sim-

ilar editing process, but use an image warping technique based on 2D mesh deformation to render output stereoscopic images. Didyk et al. [2013] used the phase-based motion magnification technique [Wadhwa et al., 2013] to modify and retarget disparities of stereo content to create pre-filtered, multi-view output for autostereoscopic displays. Focusing on perceptual issues, Didyk et al. [2012b; 2012a; 2011] proposed remapping operators that minimize the discomfort perceived by the human visual system. The modified disparity is rendered back to stereo views using the technique based on image decomposition.

As for the single view methods, all of these methods use smooth 2D warping of a stereoscopic image pair or image inpainting techniques to deal with modified disparities, and thus they are prone to bend salient scene structures such as straight lines. Furthermore, other visually relevant cues such as disocclusions cannot be handled by these method. They are therefore restricted with respect to the amount of remapping that can be achieved without producing noticeable visual distortions, and do not allow for per-pixel control over disparities. Moreover, they cannot easily generalize to more than two input views. Our approach inherently benefits from richer scene information, and is fundamentally different from the aforementioned methods: it selects actual light rays from an input light field in order to achieve per-pixel disparity control, instead of using image deformations or inpainting.

3

Acquisition

This chapter formally defines the light field and introduces the notations used throughout the thesis. Common parameterization schemes of light fields are presented as well as the their acquisition methods widely used in practice.

3.1 Light Field Parameterization

A light field represents the light transport at all positions in space for all directions at all time. As the light ray is the elementary entity that transports light and the radiance the unit for the amount of transport, a light field is a radiance function of a ray, and its parameterization schemes directly follow those of rays. In the following, common parameterization schemes are presented in the order of decreasing dimensions.

3.1.1 The Plenoptic Function

When Adelson and Bergen [1991] proposed the plenoptic function, they defined it as a function of a 7-dimensional domain and a scalar range:

$$P(x, y, z, \theta, \phi, \lambda, t). \tag{3.1}$$

Its domain includes a 3D position (x, y, z), a direction at that position as a polar angle (θ, ϕ), a wavelength λ, and a point in time t; its range is a real-valued spectral radiance. Since it is unrealistic to sample the function directly because of its high

dimensionality and measurement difficulties, simplification is further made in many practical uses.

Often omitted in practice are the wavelength λ and the time t. In most representations widely used, the radiance is sampled at three wavelength bands, often coinciding with red, green, and blue according to three types of photoreceptors in the human visual system, and stored as a tuple of three real numbers. This drops the wavelength, but turns the scalar function to a vector valued function in $\mathbb{R}^3$. The time is often treated as a fixed parameter and one deals with a snapshot of the light field captured at a particular time, further discarding one dimension. This lets us drop two parameters from the full plenoptic function, leaving five geometric parameters only—the position and the direction.

These five parameterize a ray in 3D space and form the so called *ray space*, where a line (ray) is represented as a 5D point. In order to capture the complete visual information of a 3D scene, one has to sample this 5D space as densely as required for the postulated application scenario. In practice, it is not easy to place the sensor, usually a camera, to measure the radiance in, e.g., concave parts of the scene without blocking natural illumination. Further, the 2D rectilinear image of conventional cameras has to be projected onto the surface of a spherical parameterization for directional sampling, but so does only with large distortions. Together with its limited field of view, it requires that a specialized, omnidirectional sensor probe be designed and the projection be carefully handled. For those reasons, a few further assumptions are made in most practical parameterizations: sensors are positioned on a 2D convex manifold in 3D space, directions are sampled on 2D Cartesian coordinates instead of spherical coordinates, and so on.

3.1.2 4D Light Fields

Among the most widely used parameterization schemes is the *two-plane parameterization*, where one is interested in only the rays passing though one plane π_0, followed by the other π_1, and any such ray is parameterized by the coordinates of its two intersections (s, t) and (u, v) with two planes π_0 and π_1, respectively; see Figure 3.1. With this parameterization, one can only capture the radiance in free space outside of a convex region, and may need to capture several light fields to cover the entire exterior of the convex region. Further, the radiance of a ray is assumed constant in free space along its progress. In practice such assumptions do not impose much limitation, since it is rarely realistic to place a sufficiently large number of sensors at 3D locations in a considered scene to measure the radiance, and in many scenes there is little interaction between rays and the medium—usually the air. As it requires four parameters to describe a ray, the light field parameterized accordingly is a 4D function $L_4 : \mathbb{R}^4 \to \mathbb{R}^3$ with the radiance $\mathbf{r} \in \mathbb{R}^3$

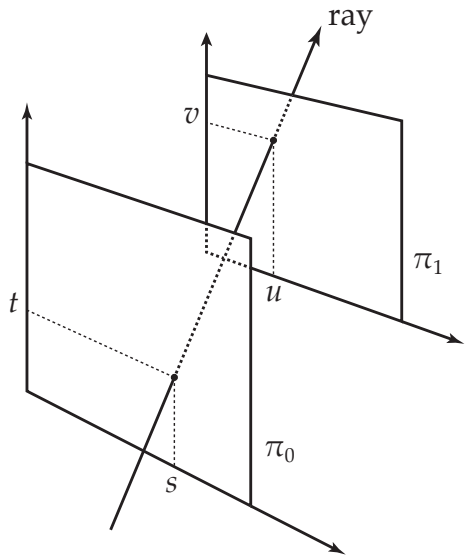


Figure 3.1: Two plane parameterization of a ray. The ray is parameterized by the two intersections (s, t) and (u, v) with planes π_0 and π_1, respectively.

given as:

$$\mathbf{r} = L_4(u, v, s, t). \tag{3.2}$$

Since its first use in Levoy and Hanrahan [1996] and Gortler et al. [1996], the two-plane parameterization is predominantly used in most literature. Its wide adoption is largely due to its conceptual simplicity and, more importantly, its native compatibility to the typical 2D rectilinear alignment of acquisition devices like the micro-lenses or camera arrays, and the 2D regular pixel arrangement of most imaging sensors.

3.1.3 3D Light Fields and EPIs

If we fix one of the two coordinates on π_0, say t, so that π_0 reduces to a line, the ray space of the resulting light fields will span the u, v, and s dimensions of the original ray space. We call a such parameterized light field a *3D light field*. A 3D light field can be denoted as a function $L_3 : \mathbb{R}^3 \to \mathbb{R}^3$. The radiance $\mathbf{r} \in \mathbb{R}^3$ of a light ray is given as

$$\mathbf{r} = L_3(u, v, s), \tag{3.3}$$

where s describes the 1D ray origin and (u, v) represent the 2D ray direction.

Several 2D slices of a light field have been known already, but as different names. A *us*-slice is obtained by reducing one dimension, v, also from π_1. Often called a *flatland light field*, it represents a light field of a hypothetical height-less world,

where the light field is parameterized by two lines instead of planes. In this hypothetical world, a light field may be obtained using (1D) pinhole cameras aligned on a line. There, every pair of two pinhole cameras forms epipolar geometry, and in particular, a special case of epipolar geometry where the epipolar plane of any pair overlaps entirely with any other pair's. Further, any (2D) point in the flat world seen from one camera has the same epipolar line against any other cameras. For these reasons, such slices were named as *epipolar-plane images* (EPI) by Bolles et al. [1987]. We denote an EPI as $E_v : \mathbb{R}^2 \to \mathbb{R}^3$, with radiance

$$\mathbf{r} = E_v(u, s) \tag{3.4}$$

of a ray at position (u, s) and fixed parameter v. We revisit the 3D light field and EPI in Chapter 4 and study them more thoroughly.

A uv slice fixing s and t is simply a perspective pinhole image $I_{s,t}(u, v)$. A vs- or ut-slice has been known as a push-broom image. Push-broom images can be obtained using a line-sensor sweeping the scene in the direction orthogonal to its linear sensor alignment, and have been widely used in satellite imaging [Gupta and Hartley, 1997] and manufacturing inspection on belt conveyor systems [Soukup et al., 2014].

3.1.4 Notational Conventions

In many works in computer graphics, the phrase plenoptic function refers to the full 7D function, or less frequently the 5D function of geometric parameters, while the light field and the Lumigraph specifically mean the 4D function. In this thesis, we use the phrase light field exclusively regardless of the dimensionality, and specify the dimension when necessary. Our notation for 4D light fields coincides with that of Gortler et al. [1996], while (u, v) in our notation is (s, t) of Levoy and Hanrahan [1996] and vice versa. For the rest of the thesis, we omit the subscript 3 or 4 from the notation for a light field L. However, the dimension of the light field should be obvious from the context. In analogy to image-pixels, we often use in this thesis the term *EPI-pixel* (u, s) instead of the term ray at (u, s) for disambiguation. Much of our discussion considers individual EPIs with parameter v fixed, and we often omit the subscript v from an EPI E for notational simplicity.

Often, the terms "spatial" and "angular" are used in a confusing way. We use "spatial" to denote what is related to the uv-plane (π_1), and "angular" to the st-plane (π_0), which seems more conventional in literature. This usage of words, however, admits of ambiguity as a uv coordinate actually determines the direction of a ray and an st coordinate its origin. We use the terms "directional" and "positional" for disambiguation. We use either of the two sets of terms whenever appropriate, but try not to mix them.

Unless stated otherwise, we assume linearized RGB color space, where a sampled radiance $\mathbf{r}$ is represented as a 3-vector (r, g, b) in the unit cube $[0, 1]^3 \subset \mathbb{R}^3$. A parenthesized list of scalars, e.g., (u, v), is used to denote a column vector $[u\ v]^\mathsf{T}$, when there is no ambiguity.

3.2 Light Field Acquisition

While numerous methods are available for light field acquisition, they are largely categorized into two classes, structured setups and unstructured setups, and the structured setups may be further divided into two categories depending on the types of optics used. This section describes only the most common acquisition setups.

3.2.1 Camera Arrays and Gantries

A straightforward sampling approach of a light field parameterized by two planes is to place cameras on one plane facing towards the other. In the acquisition setups using a camera array or a camera gantry, these camera positions are populated by multiple cameras or traversed sequentially by a single camera operated by, e.g., a robotic arm or stage. Each camera's center of projection is usually placed on a 2D regular grid on π_0, such that the two-dimensional coordinate (s, t) of the camera's position on π_0 directly relate to two of the four light field parameters. The cameras are so oriented that the common virtual image plane matches π_1 and the image coordinates (u, v) coincide with the rest of light field parameters. Thus any pixel in the collection of images represents a (box-filtered) sample radiance of the ray that corresponds to a 4D point (u, v, s, t) in the ray space of the light field. In practice, the captured images are rectified to compensate for alignment errors. Rectification can be done by estimating a 2D homography between each image and the reference coordinate frame of π_1 and warping it according to the estimated homography [Levoy and Hanrahan, 1996], but full 3D camera pose estimation can also be used; see Section 3.3.

A special case includes when the coordinate system of π_1 is defined local to each camera position $(s, t) \in \pi_0$. This happens when the principal axes of all cameras are oriented the same and perpendicular to π_0. Mathematically, this is when π_1 is placed at infinity. In this case, the (u, v) coordinates alone denote directions, the two-plane parameterization reducing to the plenoptic function's position–direction representation with the position defined on a 2D manifold in 3D space.

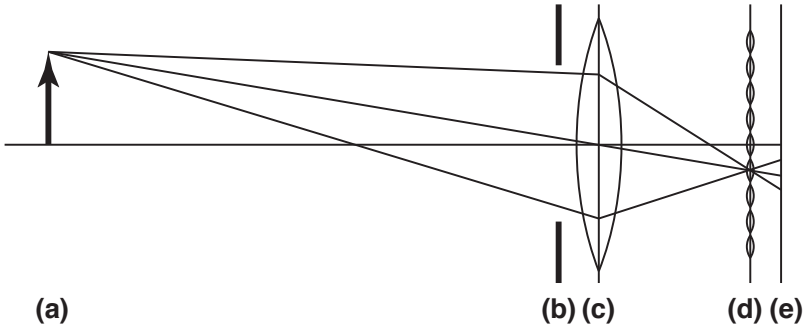


Figure 3.2: Schematic of a light field camera (not drawn to scale): **(a)** an object, **(b)** camera (main lens) aperture, **(c)** main lens, **(d)** micro-lens array, and **(e)** image sensor.

3.2.2 Light Field Cameras

While their individual optics may vary, light field cameras usually feature a sheet of micro-lenses aligned on a regular 2D grid (Figure 3.2d), between the main lens (Figure 3.2c) and the image sensor (Figure 3.2e) compared to conventional cameras. These micro-lenses cover a small portion of sensor pixels and create the focused image of the main lens aperture from different viewing angles, behaving like micro-cameras sitting atop the sensor plane. Each micro-lens splits the converged light rays based on their directions, allowing the array of pixels underneath it to record the individual rays from different sub-areas of the main lens. The optics is usually designed such that each micro-lens records the sharpest image of the aperture of the main lens and covers as many pixels underneath it as possible while not producing overlap with another [Adelson and Wang, 1992; Ng et al., 2005].

Any light ray entering the camera passes through the main lens and a micro-lens, intersecting a sensor pixel in the end. Letting (u, v) be the coordinate of the micro-lens intersecting the ray within the 2D regular arrangement, and (s, t) be the coordinate of intersection within the aperture of the main lens, the ray



Figure 3.3: Light field cameras. These show the first and second generation light field cameras from Lytro, © 2015 Lytro, Inc.

can be parameterized by these four parameters. The pixel where this ray ends up samples the integral of the radiance over a small 4D box around (u, v, s, t) in the ray space, spanned by the micro-lens and the pixel's conjugate area in the main lens aperture [Ng et al., 2005]. Thus, the resulting 2D image from the sensor samples the (box-filtered) 4D light field inside the camera. Related to the two-plane parameterization, the main lens corresponds to π_0 and the micro-lens array to π_1. Figure 3.3 shows two light field cameras currently on the market.

3.2.3 Unstructured Light Fields

An unstructured light field includes a sequence of images, each with its associated camera parameters. The camera parameters relate a 3D point at $\mathbf{x} = (x, y, z)$ and its image space coordinates $\mathbf{u} = (u, v)$ projected by a particular camera. They include the intrinsic and extrinsic parameters: the former abstracts the camera's imaging process while the latter represents the 3D pose of the camera. Although these images can be resampled to a regular structure closely corresponding to, e.g., a 4D light field [Gortler et al., 1996], often rays of interest defined in 3D space are looked up directly from the images [Davis et al., 2012]. The camera parameters are used in both cases.

A simplified camera model based on the pinhole camera is provided here. We assume the camera optics are free from non-linear distortions such as radial distortions and tangential distortions, which can be removed separately as a pre-processing step. Let $I(\mathbf{u})$ denote a pixel of image I at 2D coordinate $\mathbf{u} \in \Omega_I$, the domain of image I, which is usually a rectangular subset of the camera's image plane.

The camera intrinsics can be summarized in a single projection matrix:

$$K = \begin{bmatrix} f_\mathrm{m}/w_\mathrm{m} & f_\mathrm{m}/w_\mathrm{m} \cdot \cot\theta_\mathrm{axis} & u_0 & 0 \\ 0 & f_\mathrm{m}/h_\mathrm{m} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3.5}$$

where f_m is the camera focal length, w_m and h_m are the physical width and height of a pixel, all three in meters, θ_axis measures the angle between the two image plane axes, and (u_0, v_0) is the image space coordinate of the principal point.

By further assuming that pixels are square and the principal axis intersects the origin of the image plane whose two axes meet at a right angle, the matrix simplifies to a function of the focal length f measured in pixels:

$$K = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.6}$$

The extrinsic parameters encode the 3D pose of a camera, and define the geometry between cameras with respect to a global reference frame. A camera's 3D pose is determined by the position of its center of projection and the orientation of the principal axis, and can be represented by an affine transformation in 3-space:

$$M = \left[\begin{array}{ccc|c} & R & & \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{3.7}$$

where the 3×3 rotation matrix R represents the camera's orientation, and the translation vector $\mathbf{t} = -R^{\mathsf{T}}\mathbf{c}$ with the 3D position of the camera $\mathbf{c}$.

The extrinsic matrix M transforms a 3D point in the global reference frame to the camera's coordinate frame. Let us denote the homogeneous coordinates with tilde, e.g., $\tilde{\mathbf{x}} = (x, y, z, 1)$. Then the matrix M transforms the coordinate system so that

$$\tilde{\mathbf{x}}_c = M\tilde{\mathbf{x}} \tag{3.8}$$

is defined in the camera's coordinate system. The intrinsic matrix K further transforms any point in the camera coordinate system to the 2D image coordinate system:

$$\tilde{\mathbf{u}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx_c/z_c \\ fy_c/z_c \\ 1 \end{bmatrix} \sim \begin{bmatrix} fx_c \\ fy_c \\ z_c \end{bmatrix} = K\tilde{\mathbf{x}}_c, \tag{3.9}$$

so that $\mathbf{u} = (u, v)$ indicates the pixel coordinates of the imaged 3D point $\mathbf{x}_c$.

By chaining the two matrix multiplications, we relate a 3D point imaged by a camera to its pixel location in the image:

$$\tilde{\mathbf{u}} \sim K\tilde{\mathbf{x}}_c = KM\tilde{\mathbf{x}} =: P\tilde{\mathbf{x}}, \tag{3.10}$$

where the camera matrix P is defined as

$$P = KM. \tag{3.11}$$

Thus, to look up the radiance of a ray passing through a 3D point $\mathbf{x}$ and imaged by some camera i whose associated camera matrix is P_i, one has to find the intersection $\mathbf{u}$ of the ray on the image plane of camera i by Equation 3.10 and then sample the radiance from the image, i.e.:

$$\mathbf{r} = I_i(\mathbf{u}) \tag{3.12}$$

if $\mathbf{u} \in \Omega_{I_i}$, where $\mathbf{u}$ is obtained from $\tilde{\mathbf{u}}$ by dividing each component by the last component and taking the first two (see Equation 3.9).

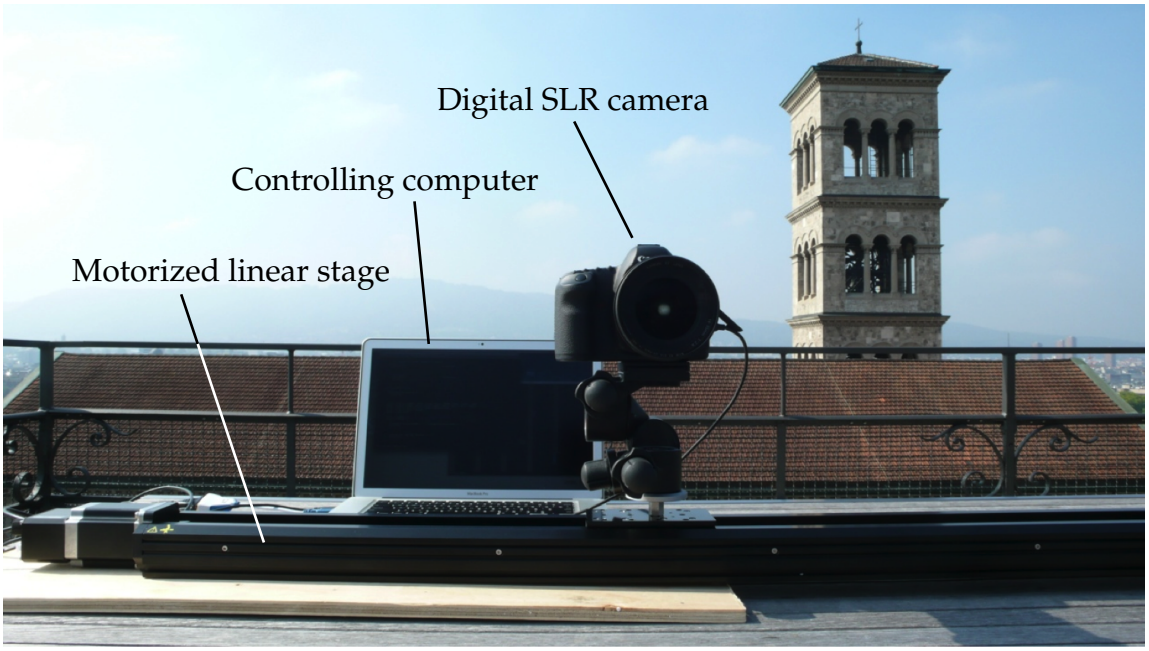


Figure 3.4: Our acquisition setup using a digital SLR camera translated by a motorized linear stage. Both the camera and the linear stage are controlled remotely from a computer.

3.3 Capture and Calibration

We primarily use a motorized linear stage to capture light fields in addition to unstructured, hand-held capture. The light fields captured using a linear stage have only one-dimensional angular variation, compared to those that are parameterized by two planes. The camera plane is replaced with a line, where the camera is located with uniform spacing. In the case of hand-held capture, images are captured at arbitrary locations and orientations and the accurate locations and orientations, i.e., 3D poses, are estimated using structure-from-motion.

3.3.1 Capture Using a Linear Stage

We capture 3D light fields by mounting a consumer digital SLR camera on a motorized linear stage. The camera is a Canon EOS 5D Mark II with a 50 mm lens, with which we capture images at various resolutions up to 5616×3744 pixels, which feature about 21 megapixels (MP). The linear stage is a Zaber T-LST1500D that is 1.5 meter long and can be controlled from a computer to obtain an accurate spacing of camera positions. We typically capture 100 images of a scene with uniform spacing between camera positions. The spacing ranges from 2 mm to 15 mm. Chapter 5 discusses the optimal number of images and amount of spacing.

The described setup works well in practice for capturing high spatio-angular resolution light fields: it is cheaper and easier to handle than a full array of cameras, while yielding much higher spatial and angular resolutions than single light field cameras based on micro-lens arrays or coded aperture. A typical capture session takes about 2 minutes, because for every picture we first move the camera, stop, take the picture, and move again to ensure accurate spacing as well as to avoid motion blur during capture. With a continuously moving setup under the illumination bright enough, the acquisition time can be reduced to a few seconds using video capture. While the spatial resolution of videos is lower than that of still images in the camera we use, one can capture light fields with a very high angular resolution thanks to a higher framerate. We use such captured light fields when evaluating our sampling model in Chapter 5. The camera is driven in a manual mode, so that all parameters including focal length, exposure, and white balance remain the same during each capture session.

3.3.2 Hand-Held Capture

For the geometry reconstruction addressed in Chapter 4 unstructured light fields are also used as input. We use the same camera as for the capture using the linear stage described in Section 3.3.1, and also fix all camera settings during each capture. A single capture includes a various number of images, mostly ranging between 50 to 100. The camera trajectory is often along a curved line roughly parallel to the ground, attributed to the ease of manual capture.

3.3.3 Post-processing of Captured Images

All the captured images are corrected for gamma in case a non-linear encoding is used, so that the pixel intensity has the linear response to the radiance. To closely approximate a pinhole camera model, we use relatively large f-numbers (usually around $f/8$) and correct the captured images for non-linear lens distortions using PTLens[1].

To compensate for possible mechanical inaccuracies of the motorized linear stage, we estimate the camera poses using Voodoo camera tracker[2], compute the least orthogonal distance line from all camera centers as a baseline, and then rectify all images with respect to this baseline [Fusiello et al., 2000].

For hand-held capture, the captured images are not rectified. Instead, the estimated camera parameters are directly used to look up the rays required, as presented in

[1] http://www.epaperpress.com/ptlens/
[2] http://www.digilab.uni-hannover.de/docs/manual.html

Section 3.2.3. Again, the camera parameters are estimated using a structure-from-motion technique. However, we use VisualSFM[3] for this, since Voodoo camera tracker is designed for near-linear camera paths and VisualSFM often outperforms it for non-linear, unstructured camera paths. We run it with the radial distortion coefficient excluded from the estimation. After successful execution, VisualSFM returns the focal length in pixels, the camera rotation as a quaternion, and the camera 3D position, which are used to construct the camera intrinsic and extrinsic matrices in Equations 3.6 and 3.7.

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

4

Geometry Reconstruction

This chapter describes a method for scene reconstruction of complex, detailed environments from light fields. Densely sampled light fields on the order of 10^9 light rays allow us to capture the real world in unparalleled detail, but efficiently processing this amount of data to generate an equally detailed reconstruction represents a significant challenge to existing algorithms. We propose an algorithm that leverages coherence in massive light fields by breaking with a number of established practices in image-based reconstruction. Our algorithm first computes reliable depth estimates specifically around object boundaries instead of interior regions, by operating on *individual light rays* instead of image patches. More homogeneous interior regions are then processed in a *fine-to-coarse* procedure rather than the standard coarse-to-fine approaches. At no point in our method is any form of global optimization performed. This allows our algorithm to retain precise object contours while still ensuring smooth reconstructions in less detailed areas. While the core reconstruction method handles general unstructured input, we also introduce a *sparse representation* and a *propagation scheme* for reliable depth estimates which make our algorithm particularly effective for 3D input, enabling fast and memory efficient processing of "gigaray light fields" on a standard GPU. We show dense 3D reconstructions of highly detailed scenes, enabling applications such as automatic segmentation and image-based rendering, and provide an extensive evaluation and comparison to existing image-based reconstruction techniques.

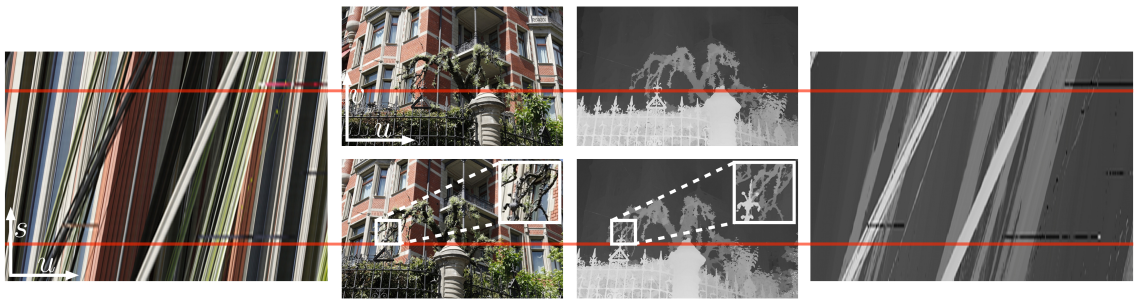


Figure 4.1: Our method reconstructs accurate depth from light fields of complex scenes. The images on the left show a 2D slice of a 3D input light field, a so called epipolar-plane image (EPI), and two out of one hundred 21 megapixel images that were used to construct the light field. Our method computes 3D depth information for all visible scene points, illustrated by the depth EPI on the right. From this representation, individual depth maps or segmentation masks for any of the input views can be extracted as well as other representations like 3D point clouds. The horizontal red lines connect corresponding scanlines in the images with their respective positions in the EPI.

4.1 Introduction

Scene reconstruction in the form of depth maps, 3D point clouds or meshes has become increasingly important for digitizing, visualizing, and archiving the real world, in the movie and game industry as well as in architecture, archaeology, arts, and many other areas. For example, in movie production considerable efforts are invested to create accurate models of the movie sets for post-production tasks such as segmentation, or integrating computer-generated and real-world content. Often, 3D models are obtained using laser scanning. However, because the sets are generally highly detailed, meticulously designed, and cluttered environments, a single laser scan suffers from a considerable amount of missing data at occlusions [Yu et al., 2001]. It is not uncommon that the manual clean-up of hundreds of merged laser scans by artists takes several days before the model can be used in production.

Compared to laser scanning, an attractive property of passive, image-based stereo techniques is their ability to create a 3D representation solely from photographs and to easily capture the scene from different viewing positions to alleviate occlusion issues. Unfortunately, despite decades of continuous research efforts, the majority of stereo algorithms seem not well suited for today's challenging applications, e.g., in movie production [Sylwan, 2010], to efficiently cope with higher and higher resolution images[1] while at the same time producing sufficiently accurate and reliable reconstructions. For specific objects like human faces stereo-based

[1]Digital cinema and broadcasting are in the process of transitioning from 2k to 4k resolution (~2 megapixels to ~9 megapixels).

techniques have matured and achieve very high reconstruction quality (e.g., Beeler et al. [2010]), but more general environments such as the detailed outdoor scene shown in Figure 4.1 remain challenging for *any* existing scanning approach.

In this thesis we follow a different strategy and revisit the concept of 3D light fields, i.e., a dense set of photographs captured along a linear path. In contrast to sparser and less structured input images, a perfectly regular, densely sampled 3D light field exhibits a very specific internal structure: every captured scene point corresponds to a linear trace in a so called epipolar-plane image (EPI), where the slope of the trace reflects the scene point's distance to the cameras (see Figure 4.1). The basic insight to leverage these structures for scene reconstruction was proposed as early as in 1987 [Bolles et al., 1987], and has been revisited repeatedly since then (see, e.g., Criminisi et al. [2005]). However, these methods do not achieve the reconstruction quality of today's highly optimized two or multi-view stereo reconstruction techniques.

With today's camera hardware it has become possible to capture truly dense 3D light fields. For example, for the results shown in Figure 4.1 we captured one hundred 21 megapixel (MP) images using a standard digital SLR camera, effectively resulting in a "two–gigaray" light field. While such data can capture an unparalleled amount of detail of a scene, it also poses a new challenge. Over many years the basic building blocks in stereo reconstruction such as patch-based correlation, edge detection and feature matching have been tailored towards optimal performance at about 1–2 MP resolution. In addition, most algorithms involve some form of global optimization in order to obtain sufficiently smooth results. As a consequence, it is often challenging to scale such approaches to significantly higher image resolution.

In this chapter we propose an algorithm that specifically leverages the properties of densely sampled, high resolution 3D light fields for reconstruction of static scenes. Unlike approaches based on patch-correlation our algorithm operates at the single pixel level, resulting in precise contours at depth discontinuities. Smooth, homogeneous image regions are handled by a hierarchical approach. However, instead of a standard coarse-to-fine estimation, we reverse this process and propose a *fine-to-coarse* algorithm that reconstructs reliable depth estimates at the highest resolution level first, and then proceeds to lower resolutions, avoiding the need for any kind of explicit global regularization. At any time the algorithm operates only on a small set of adjacent EPIs, enabling efficient GPU implementation even on light fields in the order of 10^9 rays. We further increase efficiency by propagating reliable depth estimates throughout the whole light field using a novel sparse data structure, such that the algorithm effectively computes depth maps for all input images concurrently. We also discuss how our reconstruction algorithm generalizes to 4D light fields and unstructured ones, and how it can be applied to general 3D

reconstruction problems. We demonstrate dense reconstructions of challenging, highly detailed scenes and compare to a variety of related stereo-based approaches. We also present direct applications to segmentation and novel-view synthesis.

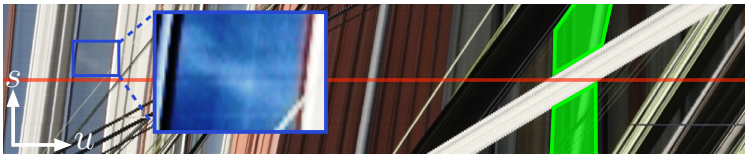
4.2 Sparse Representation

Light fields are typically constructed from a large set of images of a scene, captured at different viewing positions. A suitable representation of such data depends on a plethora of factors, including for example structured vs. unstructured capture of light fields, the targeted processing algorithms and applications, or just the sheer amount of data. Accordingly various representations have been proposed in the past [Levoy and Hanrahan, 1996; Gortler et al., 1996; Isaksen et al., 2000; Buehler et al., 2001; Davis et al., 2012]. Our main focus is on 3D light fields of very high spatio-angular resolution, i.e., light fields constructed from hundreds of high resolution 2D images with their respective optical centers distributed along a 1D line. We introduce a novel compact representation that enables efficient parallel processing without the need to keep the full input light field in memory, and that can be efficiently constructed during our depth estimation described in Section 4.3.

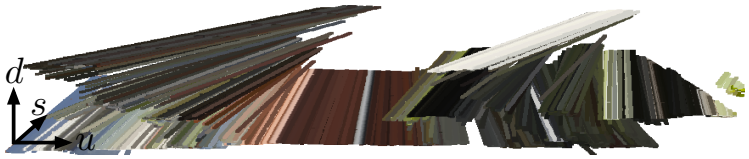
A 3D light field can be denoted as a map $L : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ with the radiance $\mathbf{r} \in \mathbb{R}^3$ of a light ray given as $\mathbf{r} = L(u, v, s)$, where s describes the 1D ray origin and (u, v) represents the 2D ray direction. While for given s, a uv-slice of this light field corresponds to an input image, denoted by I_s, a us-slice for a fixed v coordinate corresponds to an epipolar-plane image, or EPI, denoted by $E_v : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ with radiance $\mathbf{r} = E_v(u, s)$. The left half of Figure 4.1 shows two out of 100 input images and an exemplary EPI. The horizontal red lines visualize both the respective s-parameters of the two input images in the EPI as well as the v-parameter in the input images from which the EPI has been constructed. See Chapter 3 for details on notations and on how to capture 3D light fields in practice.

When the ray space of a 3D light field L is sampled densely enough, each scene point appears as a line segment in such an EPI with the slope of the line segment depending on the scene point's depth. Correspondingly, the EPIs of 3D light fields exhibit high coherence and contain very redundant information that can be utilized for a more efficient representation. Rather than storing the full EPI, we can in principle reconstruct it by knowing the parameters of those line segments. While this basic idea is well known, we propose a new representation that specifically considers two new aspects, namely *completeness* and *variation* of the represented light field.

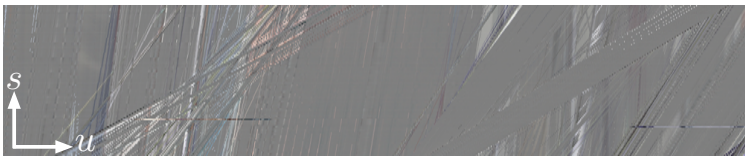
Assume we can accurately estimate the slope of line segments or, equivalently, the depth of scene points. A first idea could be to simply collect and store the line



(a) EPI E



(b) 3D visualization of Γ



(c) Delta EPI ΔE

Figure 4.2: Illustration of our sparse representation, using a cropped section from the EPI in Figure 4.1. **(a)** Concerning *completeness*, consider the region shaded in green on the right. It is occluded by the white structure and thus propagating color values from only the central view, marked by the red horizontal line, would not reconstruct the highlighted region. View-dependent *variation*, e.g., due to reflections in the building windows, is highlighted in the blue framed region. We increased color contrast in the inset for improved visibility of the color changes. Again, a reconstruction solely from the central view would not capture these effects. **(b)** 3D visualization of EPI $\widehat{E}$ reconstructed from our sparse representation Γ. **(c)** Visualization of the difference between the input EPI and our reconstructed EPI $\widehat{E}$.

segments and their color along a single horizontal line of an EPI. In principle this corresponds to storing a single input image and a depth map. A large number of captured light rays may be occluded in this particular part of the EPI, hence *completeness* of the representation would be compromised. In addition, scene points may change their color along their corresponding line segment due to specularities or other view dependent effects. Hence the above representation would not capture *variation* in the light field. See Figure 4.2a for a visualization of both effects.

Our strategy for representing 3D light field data addresses these two issues. First, we sample and store a set Γ of line segments originating at various locations (u, s) in the input EPI E, until the whole EPI is completely represented and redundancy is eliminated to the extent possible. Second, we store a difference EPI ΔE that accounts for variations in the light field. More specifically, the slope m of a line

segment associated with a scene point at distance z is given by

$$m = \frac{1}{d} = \frac{z}{f \cdot b} \ , \tag{4.1}$$

where d is the image space disparity defined for a pair of images captured at adjacent positions or, equivalently, the displacement between two adjacent horizontal lines in an EPI, f is the camera focal length in pixels and b is the metric distance between each adjacent pair of imaging positions. Correspondingly an EPI line segment can be compactly described by a tuple

$$\mathbf{p} = (m, u, s, \mathbf{r}^{\mathsf{T}}), \tag{4.2}$$

where $\mathbf{r}$ is the average color of the scene point in the EPI. Γ is simply the set of all tuples $\mathbf{p}$. The actual scheme of how we collect line segments $\mathbf{p}$ is part of the depth computation described in the following section.

From Γ, a reconstructed EPI $\widehat{E}$ can be generated by rendering the line segments in the order of decreasing slopes, i.e., render the scene points from back to front. See Figure 4.2b for a 3D visualization of the full representation Γ. Hence, for efficient EPI reconstruction, Γ is stored as an ordered list of tuples in the order of decreasing slopes. The difference $\Delta E = E - \widehat{E}$ of the input E and the reconstruction $\widehat{E}$ captures the remaining variation and detail information in the light field, such as view dependent effects. This is illustrated in Figure 4.2c, where 50 % gray corresponds to zero reconstruction error. Note a high value of ΔE for the specularities and at inaccurate slope estimates.

Both Γ and ΔE compactly store all relevant information that is necessary to reconstruct the full 3D light field as well as extract an arbitrary input image with a corresponding depth map, or a full 3D point cloud. As an example, for the EPI in Figure 4.2, ~277 k EPI-pixels are reduced to ~15 k tuples (about 5.7 %). Plain storage of the full tuple information without any further compression already results in a reduction to 21 % compared to the RGB EPI. As discussed above various alternatives exist to store a coherent light field. A main benefit of our representation is its consistency with our algorithm for depth computation, enabling compact representation and efficient parallel computation as described in the next section.

4.3 Depth Estimation

Constructing Γ amounts to computing the line slopes at the EPI-pixels, i.e., estimating the depth of scene points. As mentioned before the ray coherence of a dense 3D light field allows our algorithm to operate on individual EPI-pixels instead of having to consider larger pixel-neighborhoods like most stereo approaches. As a

consequence it performs especially well at depth discontinuities and reproduces precise object silhouettes due to the color contrast in these regions. This property is key to our *fine-to-coarse* depth estimation strategy: we estimate depth first at edges in the EPI at the highest resolution, propagate this information throughout the EPI, and then proceed to successively coarser EPI resolutions. In contrast to classic coarse-to-fine schemes, this allows us to preserve sharp depth discontinuities at object silhouettes, while also estimating accurate depth in homogeneous regions. Additionally, our strategy increases computational efficiency by restricting computations to small fractions of the high resolution input.

Starting at the full resolution of an EPI E, the first step consists of efficiently identifying regions where the depth estimation is expected to perform well. To this end we introduce a fast *edge confidence* measure C_e that is computed on the EPI. The algorithm then generates depth estimates for EPI-pixels with a high edge confidence. This is done by testing various discrete depth values d from the set $\mathcal{H}$ of hypotheses and picking the one that leads to the highest color density of sampled EPI-pixels. The density estimation is further leveraged to improve the initial confidence towards a refined *depth confidence* C_d, which provides a good indicator for the reliability of a particular depth estimate. All EPI-pixels with a high reliability are stored as tuples in Γ and propagated throughout the EPI. This process of depth estimation and propagation is iterated until all EPI-pixels with a high edge confidence C_e have been processed.

At this point all confident, i.e., sufficiently detailed regions at the current resolution level of the EPI E have a reliable depth value assigned, while the depth in more homogeneous regions is yet unknown. Our fine-to-coarse approach then downsamples E to a coarser resolution and starts over with the above procedure, computing edge confidence for yet unprocessed parts of the EPI and so forth. This procedure is continued until a depth value is assigned to every EPI-pixel, i.e., the line segment tuples in Γ reconstruct the complete light field.

4.3.1 Edge Confidence

As the edge confidence measure C_e is intended to be a fast test for which parts of the EPI a depth estimate seems promising, we define it as a simple difference measure

$$C_e(u,s) = \frac{1}{|\mathcal{N}(u,s)|} \sum_{u' \in \mathcal{N}(u,s)} \|E(u,s) - E(u',s)\|_2 \,, \tag{4.3}$$

where $\mathcal{N}(u,s)$ is a 1D window in the EPI E around the pixel (u,s). The size of this neighborhood can be small (9 pixels in our experiments) as it is supposed to measure only the local color variation.

C_e is then thresholded (with a value of 0.02), resulting in a binary confidence mask M_e, visualized as red pixels in Figure 4.5c–e. In order to remove spurious isolated regions, we apply a morphological opening operator to the mask. During the following depth computation this binary mask will be used to prevent the computation of depth estimates at ambiguous EPI-pixels and hence speed up the computation without sacrificing accuracy.

4.3.2 Depth Computation

Next our algorithm computes depth estimates for EPI-pixels in E marked as confident in M_e. For simpler parallelization on a GPU we perform this computation per scanline in the EPI, i.e., we select a fixed parameter $\hat{s}$ and compute a depth estimate for all $E(u, \hat{s})$ with $M_e(u, \hat{s}) = 1$. As discussed in Section 4.2, initially we select $\hat{s}$ as the horizontal centerline of E, as this generally allows us to compute a large fraction of the line segments visible in the EPI.

Following Equation 4.1 we try to assign a depth z, or equivalently a disparity d, to each EPI-pixel $(u, \hat{s})$. For a hypothetical disparity $d \in \mathcal{H}$, the set $\mathcal{R}$ of radiances or colors of these EPI-pixels is sampled as

$$\mathcal{R}(u, d) = \{E(u + (\hat{s} - s)\, d,\ s) \mid s = 1, \ldots, n\}, \tag{4.4}$$

where n corresponds to the number of views in the light field. From the density of radiance values in $\mathcal{R}(u, d)$ a depth score $S(u, d)$ is computed in linearized RGB color space. The assumption here is that the scene is essentially Lambertian, i.e., a set $\mathcal{R}$ is likely to represent an actual scene point if the radiance samples are densely positioned in the underlying color space. Due to the high number of available samples in a dense light field our measure is very robust to outliers and hence implicitly handles occlusions. As we show in our results it is even robust to inconsistencies such as moving elements. We also experimented with other color spaces such as Lab and HSV with the hue angle represented by its sine and cosine. However, we could not find a significant difference.

We compute the density efficiently using iterations of a modified Parzen window estimation [Duda et al., 1995] with an Epanechnikov kernel, and define the initial depth score as

$$S(u, d) = \frac{1}{|\mathcal{R}(u, d)|} \sum_{\mathbf{r} \in \mathcal{R}(u,d)} K\left(\mathbf{r} - \bar{\mathbf{r}}\right), \tag{4.5}$$

where $\bar{\mathbf{r}} = E(u, \hat{s})$ is the radiance value at the currently processed EPI-pixel, and the kernel

$$K(\mathbf{x}) = \begin{cases} 1 - \left\| \frac{\mathbf{x}}{h} \right\|_2^2 & \text{if } \left\| \frac{\mathbf{x}}{h} \right\|_2 \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

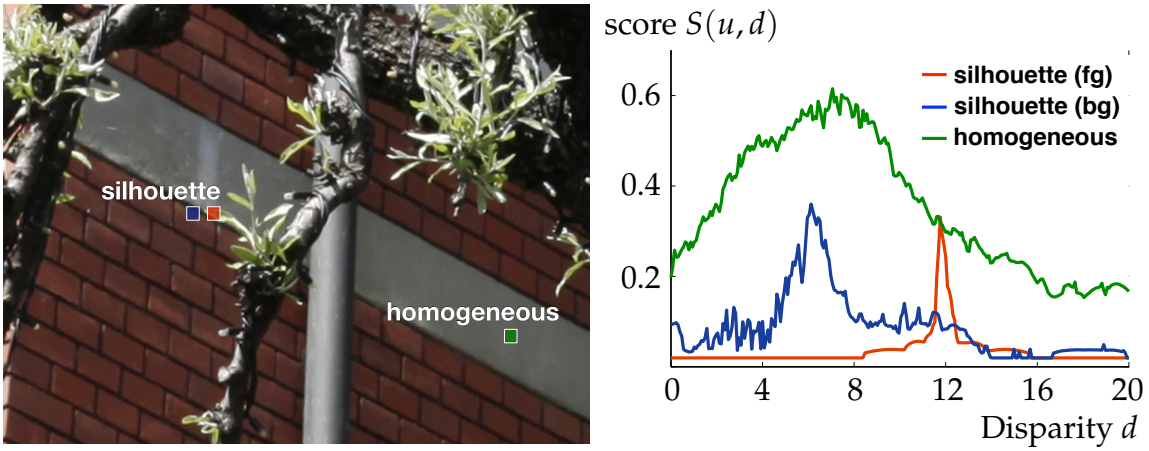


Figure 4.3: At high image resolutions silhouette pixels result in a clear peak with a distinctive score profile whereas homogeneous regions lead to more flat and ambiguous scores. On coarser resolutions the scores in homogeneous regions become more distinct, which motivates our fine-to-coarse estimation.

The bandwidth parameter was set to $h = 0.02$ in our experiments. Gaussian or other bell-shaped kernels also work well, but the chosen kernel is cheaper to compute. For a rather noise-free EPI this initial depth score is sufficient. To reduce the influence of noisy radiance measurements we borrow ideas from the mean-shift algorithm [Comaniciu and Meer, 2002] by computing an iteratively updated radiance mean

$$\bar{\mathbf{r}} \leftarrow \frac{\sum_{\mathbf{r}\in\mathcal{R}} K(\mathbf{r}-\bar{\mathbf{r}})\mathbf{r}}{\sum_{\mathbf{r}\in\mathcal{R}} K(\mathbf{r}-\bar{\mathbf{r}})} \tag{4.7}$$

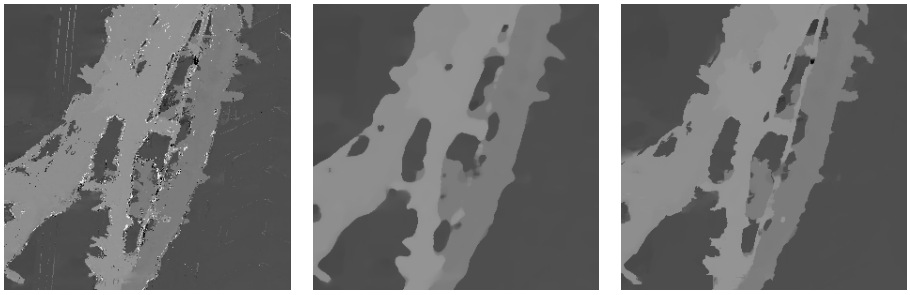
before computing Equation 4.5. Regarding the efficiency of this approach it is important to note that a full mean-shift clustering process or even just running the above mean-shift steps to convergence is counter-productive, as it significantly increases the computational complexity, in particular on a GPU due to the required branching and possibly different control flow. The main purpose, i.e., robustness to noise, is achieved already after a few iterations, hence the algorithm performs a constant number of 10 iterations for all results shown in this thesis.

For each EPI-pixel $(u, \hat{s})$ we compute scores $S(u,d)$ for the whole range of admissible disparities d, and assign the disparity with the highest score as the pixel's depth estimate

$$D(u,\hat{s}) = \operatorname*{argmax}_d S(u,d). \tag{4.8}$$

In addition we also compute the refined confidence C_d as a measure for the reliability of a depth estimate. C_d combines the edge confidence C_e with the difference between the maximum score $S_{\max}(u)$ and the average score $\bar{S}(u)$:

$$C_d(u,\hat{s}) = C_e(u,\hat{s})\,|S_{\max}(u) - \bar{S}(u)|, \tag{4.9}$$



(a) No median **(b)** Standard median **(c)** Bilateral median

Figure 4.4: Our proposed bilateral median filter removes speckles, while preserving fine details like the thin vertical string in the middle.

where

$$S_{\max}(u) = \max_d S(u,d) \tag{4.10}$$

and

$$\bar{S}(u) = \frac{1}{|\mathcal{H}|} \sum_{d \in \mathcal{H}} S(u,d). \tag{4.11}$$

The refined confidence measure C_d is meaningful as it combines two complementary measures. For instance, noisy regions of an EPI would result in a high edge-confidence C_e, while a clear maximum $S_{\max}$ is not available. Similarly, ambiguous homogeneous regions in an EPI, where C_e is low, can produce a strong, but insufficiently unique $S_{\max}$; see Figure 4.3.

In order to eliminate the influence of outliers that might have survived the density estimation process, we apply a median filter on the computed depths. However, we observed that a straightforward median filter compromises the precise localization of silhouettes. We therefore use a bilateral median filter that preserves the localization of depth discontinuities by leveraging information from the radiance values of nearby EPIs. This is implemented by replacing the depth estimate $D_v(u,\hat{s})$ by the median value of the set

$$\begin{aligned} \{\, D_{v'}(u',\hat{s}) \;\mid\; & (u',v',\hat{s}) \in \mathcal{N}(u,v,\hat{s}) \;\wedge \\ & \|E_v(u,\hat{s}) - E_{v'}(u',\hat{s})\| < \varepsilon \;\wedge \\ & M_e(u',v',\hat{s}) = 1 \,\}, \end{aligned} \tag{4.12}$$

where $(u',v',\hat{s}) \in \mathcal{N}(u,v,\hat{s})$ denotes a small window over $I_{\hat{s}}$. The second condition assures that we only consider EPI-pixels of similar radiance and the last condition masks out unconfident EPI-pixels for which no depth estimation is available. In all our experiments we use a window size of 11×11 and a threshold value $\varepsilon = 0.1$. Correspondingly, we always store at most 11 EPIs during computation. The effect of this filtering step is illustrated in Figure 4.4.

4.3.3 Depth Propagation

Each confident depth estimate $D(u, \hat{s})$ with $C_d(u, \hat{s}) > \varepsilon$ is now stored as a line segment tuple $\mathbf{p} = (m, u, \hat{s}, \bar{\mathbf{r}}^\mathsf{T})$ in Γ (see Equation 4.1), where $\bar{\mathbf{r}}$ represents the mean radiance of $(u, \hat{s})$ computed in Equation 4.7. Then the depth estimate is propagated along the slope of its corresponding EPI line segment to all EPI-pixels (u', s') that have a radiance similar to the mean radiance, i.e., $\|E(u', s') - \bar{\mathbf{r}}\| < \varepsilon$ with ε having the same value as in Equation 4.12. This step amounts to a conservative visibility estimation and ensures that foreground objects in the EPI are not overwritten by background objects during the propagation.

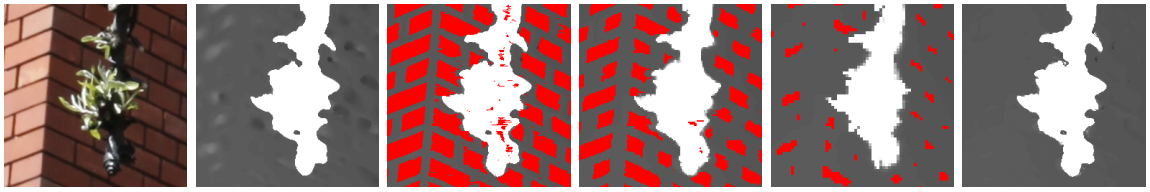
As an alternative to the above test of radiance similarities, we experimented with running the full mean shift clustering on the set $\mathcal{R}(u, d)$ and propagating the depth estimate directly to the cluster elements, but we found that our simplified density estimation and the above procedure provide similar results in a fraction of the time.

Finally, low confidence depth estimates are discarded and marked for recomputation, and all EPI-pixels with a depth estimate assigned during the propagation are masked from further computations. A new part of the EPI is selected for depth computation by setting $\hat{s}$ to the nearest s with respect to the center of the EPI that still has unprocessed pixels. The method then starts over with the radiance sampling and depth computation as described in Section 4.3.2, until all edge confident EPI-pixels at the current EPI resolution have been either processed or masked during the propagation.

4.3.4 Fine-to-Coarse Refinement

Parts of the EPI without assigned depth values are either ambiguous due to homogeneous colors (insufficient edge confidence), or have a strongly view dependent appearance (insufficient depth-confidence). However, since our method starts processing at the highest available resolution, the set Γ provides reliable reconstructions of all detailed features in the EPI and, in particular, of object silhouettes. The core idea of our fine-to-coarse strategy is now to compute depth in less detailed and less reliable regions by exploiting the regularizing effect of an iterative downsampling of the EPI. Furthermore, we enhance robustness and speed up the computation by using the previously computed confident depth estimates as depth interval bounds for the depth estimation at coarser resolutions. See Figure 4.5 for an example of our refinement strategy and note the improvement from Figure 4.5b to f at the bricks.

First the depth bounds are set for all EPI-pixels without a depth estimate. As depth



(a) Image **(b)** No fine-to-coarse **(c)** Level 0 **(d)** Level 1 **(e)** Level 2 **(f)** With fine-to-coarse

Figure 4.5: Our fine-to-coarse refinement yields reliable depth estimates also in homogeneous image regions, like the bricks. This is achieved by applying our confidence measure to detect unreliable pixels (marked in red) and estimate their depth at coarser image resolutions with the depth range bounded by estimates on the higher resolutions.

bounds, the algorithm uses the upper and lower bounds of the closest reliable depth estimates in each horizontal row of the EPI. Then the EPIs are downsampled by a factor of 0.5 along the spatial u and v-dimensions, while the resolution along the angular s-dimension is preserved. We presmooth the EPIs along the spatial dimensions using a 7×7 Gaussian filter with standard deviation $\sigma = \sqrt{0.5}$ to avoid aliasing. The required 7 EPIs are already in memory from the bilateral median filtering step (Equation 4.12).

The algorithm then starts over at the new, coarser resolution with the previously described steps, i.e., edge confidence estimation, depth estimation and propagation. EPI-pixels with reliable depth estimates computed at higher resolutions are not considered anymore but only used for deriving the above described depth bounds. This fine-to-coarse procedure is iterated through all levels of the EPI pyramid until any of the image dimensions becomes less then 10 pixels. At the coarsest level, depth estimates are assigned to all pixels regardless of the confidence measurements. The depth estimates at coarser resolution levels are then successively upsampled to the respective higher resolution levels and assigned to the corresponding higher resolution EPI-pixels without a depth estimate, until all EPI-pixels at the finest resolution level have a corresponding depth estimate. As a final step we apply a 3×3 median to remove spurious speckles.

Note that unlike other algorithms based on multi-resolution processing and global regularization, our fine-to-coarse procedure (similar in spirit to the push-pull algorithm [Gortler et al., 1996]) starts at the highest resolution level and hence preserves all details, which is generally very challenging in classical, coarse-to-fine multi-resolution approaches. Our downsampling achieves an implicit regularization for less reliable depth estimates so that all processing steps are purely local at the EPI-level. Hence, even massive light fields can be processed efficiently.

4.3.5 Extension to 4D Light Fields

In this section and the next, we extend our algorithm to input other than 3D light fields. For such input, we lose the efficiency of the EPI-based processing, but the core algorithmic steps generalize and the reconstruction quality remains.

It is straightforward to generalize our reconstruction algorithm to 4D light fields. In a regular 4D light field the camera centers are both horizontally and vertically displaced, and the additional vertical displacement is parameterized by t; see Chapter 3 for details. This leads to a 4D parametrization of rays, and the radiance of a ray is looked up as $\mathbf{r} = L(u, v, s, t)$. The ray sampling from Equation 4.4 is then extended to

$$\mathcal{R}(u, v, s, t, d) = \{L(u + (\hat{s} - s)\, d,\ v + (\hat{t} - t)\, d,\ s,\ t) \mid\ s = 1, \ldots, n,\ t = 1, \ldots, m\}, \tag{4.13}$$

where $(\hat{s}, \hat{t})$ is the considered view and m denotes the number of vertical viewing positions. This leads to sampling a 2D plane in a 4D ray space instead of the 1D line in case of 3D light fields.

4.3.6 Extension to Unstructured Light Fields

For arbitrary, unstructured input we use the camera poses estimated in the calibration phase (see Section 3.3) to determine the set of sampled rays for a depth hypothesis. More precisely, we back-project each considered pixel to 3D space in accordance to the hypothesized depth and then re-project the 3D position to the image coordinate systems of all other views to obtain the sampling positions.

Let us consider an image-space coordinate $\mathbf{u}_s = (u, v)$ at a view s and its augmented vector $\tilde{\mathbf{u}}_s = (u, v, f)$ with respect to the focal length f in pixels. Let M_s denote the camera extrinsic matrix defined as Equation 3.7, which is a 4×4 affine transformation matrix in 3-space, comprising the rotation and the translation of the camera coordinate system of the view s with respect to a reference coordinate system.

With this setting, the 3D position $\mathbf{x}_{\hat{s}}$ in the coordinate system of reference view $\hat{s}$ defined by $\mathbf{u}_{\hat{s}}$ and a hypothesized depth z is given by

$$\mathbf{x}_{\hat{s}} = \frac{z}{f}\, \tilde{\mathbf{u}}_{\hat{s}} \equiv \frac{1}{d}\, \tilde{\mathbf{u}}_{\hat{s}}, \tag{4.14}$$

where d is analogous to the image space disparity as before, but defined up to scale. The same 3D position seen from another camera at s can be computed as

$$\tilde{\mathbf{x}}_s = M_s\, M_{\hat{s}}^{-1}\, \tilde{\mathbf{x}}_{\hat{s}}, \tag{4.15}$$

where

$$\tilde{\mathbf{x}} = (\mathbf{x}^\mathsf{T}, 1) \sim (\tilde{\mathbf{u}}^\mathsf{T}, d) = (u, v, f, d) \tag{4.16}$$

denotes the homogeneous coordinate of a 3D position $\mathbf{x}$. Thus the sampled set of rays at position (u, v) in view s for a depth hypothesis d can be defined as

$$\mathcal{R}(u, v, s, d) = \{L(u', v', s) \mid M_s^{-1} [\, u' \, v' \, f \, d \,]^\mathsf{T} = M_{\hat{s}}^{-1} [\, u \, v \, f \, d \,]^\mathsf{T}, \\ s = 1, \ldots, n\}. \tag{4.17}$$

4.4 Experimental Evaluation

This section presents results and evaluations of our method, including comparisons to various state-of-the-art techniques in (multi-view) stereo. We also demonstrate exemplary applications such as segmentation and image-based rendering. Finally, we show the results of 4D and unstructured light fields.

4.4.1 Results

Using the acquisition setup presented in Section 3.3 we captured a variety of 3D light fields of challenging outdoor and indoor scenes. In Figures 4.6 and 4.7 we show example input images and corresponding depth maps. However, our algorithm computes depth for every scene point that is visible in the input images. Hence, from our internal representation we can efficiently extract depth maps for each input view, as well as generate alternative scene representations like 3D point clouds. Figures 4.6 and 4.7 additionally show 3D meshes extracted from our reconstructions. The meshes were obtained by triangulating individual depth maps and merging them into a single model. To enhance visualization we color coded vertices according to their depth (red for near vertices and blue for far). Our method faithfully reproduces fine details of complex, cluttered scenes, with precise reconstruction of object contours, performing well on homogeneous regions at the same time. These properties are highly desirable in applications such as segmentation (Figure 4.15) or novel view synthesis with moderate viewpoint changes (Figure 4.16).

Figure 4.8 demonstrates the robustness of our algorithm for different numbers of input views. We ran our experiments on a desktop PC with an Intel Core i7 3.2 GHz CPU and an NVidia GTX 680 graphics card, and tested a set of 256 depth hypotheses for every EPI-pixel in all experiments. As a baseline solution, we computed a result from 100 input views at the full 21 MP resolution and evaluated the error using normalized sum-of-absolute differences (SAD). While our algorithm benefits from a large number of input views, reasonable results can still be achieved

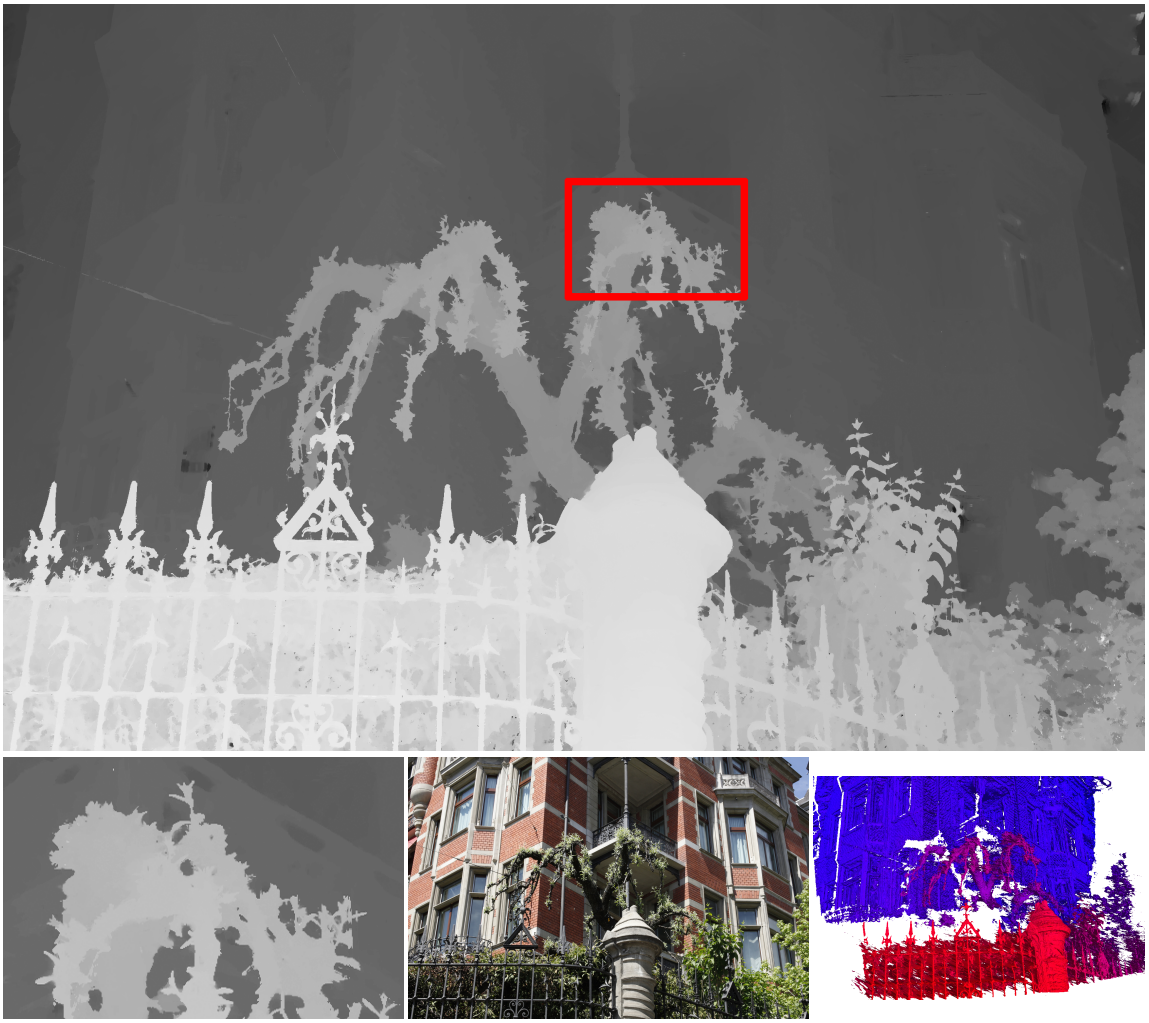
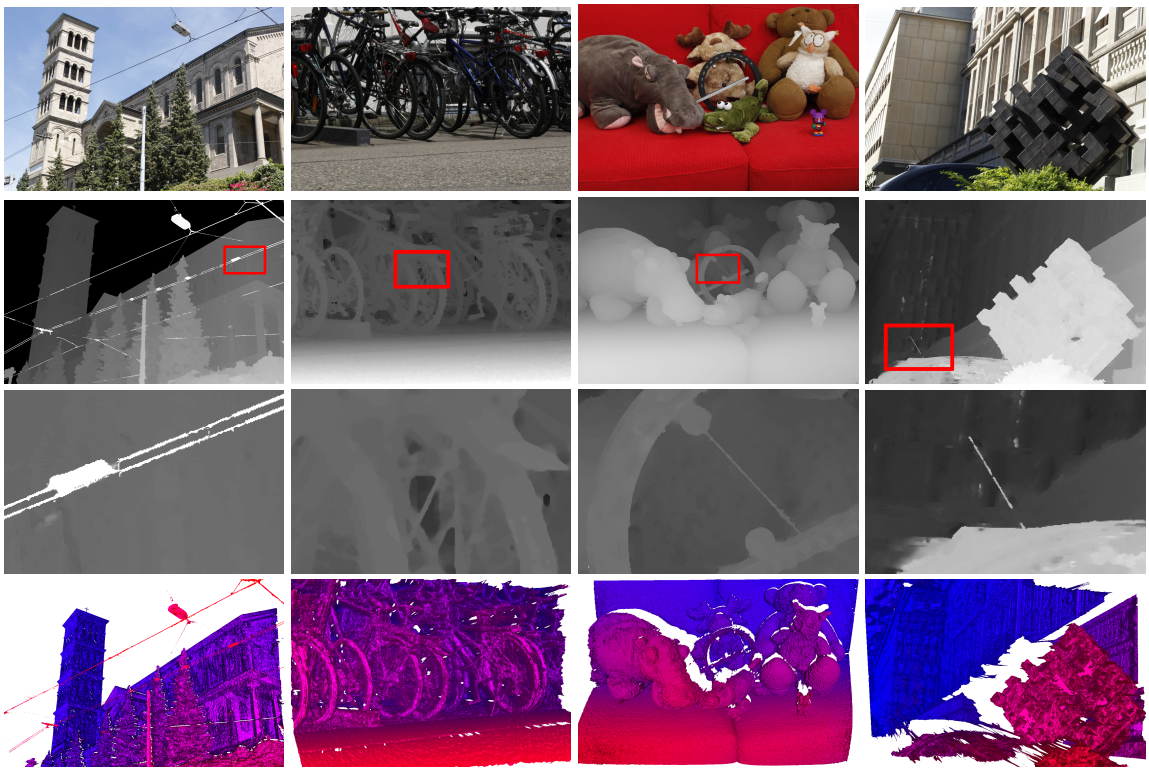


Figure 4.6: Results of the Mansion dataset, which show the reconstructed depth map and the closeup of the highlighted region along with an input view and a 3D mesh.

with only 10 input views (see Figure 4.8b). A typical runtime for a single depth map using 100 views at 21 MP resolution is about 9 minutes. With our current implementation, the full propagation to 50 views takes about 50 minutes. The linear dependence of the runtimes on the number of images is illustrated in Figure 4.8a. For example, for 10 views a single depth map requires about 1 minute.

Our method is robust against varying baseline and angular separations caused by different distances between the camera positions and the scene points. For the results shown in Figure 4.7 the angular separations range from 1.5° up to 13°. The example in Figure 4.19 captured with a hand-held camera features a considerable angular separation from 9° to 41° as well as a large baseline of about 300 meters. In addition our algorithm is robust to non-static scene elements like people moving in front of the camera or plants moving in the wind (Figure 4.9). For instance, the



(a) Church **(b)** Bikes **(c)** Couch **(d)** Statue

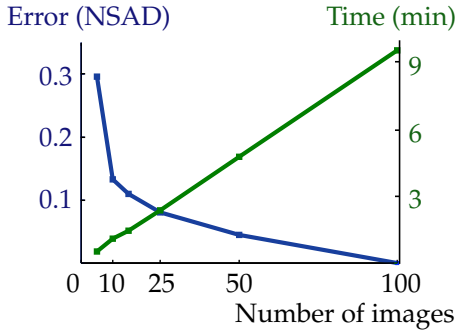
Figure 4.7: Results on various 3D light fields. *Top to bottom:* One input image, corresponding depth map, closeup of the highlighted region, and 3D mesh. For the Church dataset we used color-based segmentation to exclude the homogeneous sky as no meaningful depth can be computed there.

sparse horizontal color artifacts visible in the input EPI in Figure 4.1 are caused by people passing by during capture. The density estimation in Equation 4.5 simply regards those radiance values as outliers and still produces a consistent result from the remaining samples.

The influence of the two most relevant parameters in our method, the kernel bandwidth h and the color tolerance ε of the bilateral median, is conceptually similar to adjusting the window size in stereo methods comparing image patches. An increase of h and ε compared to our default values increases robustness to noise, whereas smaller values better preserve fine details.

4.4.2 Comparisons

We processed the Mansion data set with a number of state-of-the-art techniques in two-view and multi-view stereo, and also ran our algorithm on a number of



(a) Errors and runtimes



(b) Reconstruction from 10 views

Figure 4.8: Robustness of our method. **(a)** Reconstruction error and runtimes for varying numbers of input views. **(b)** Reconstruction from only 10 views.

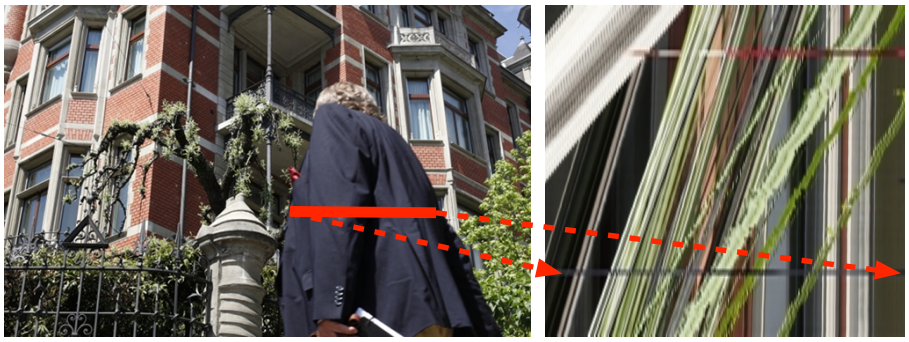
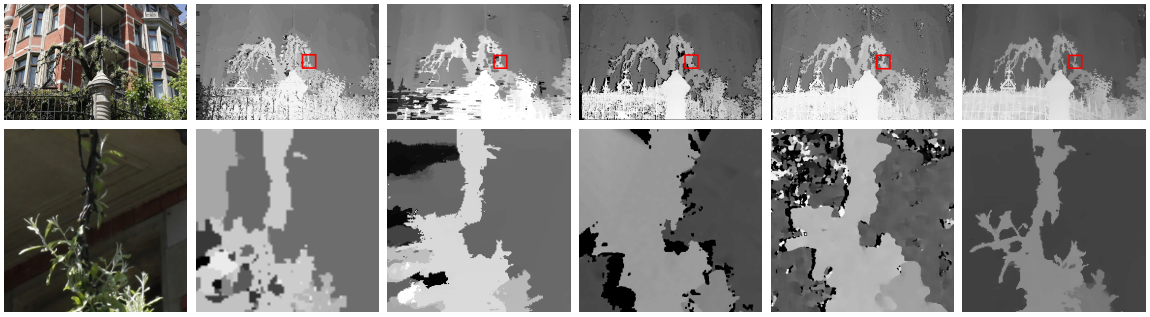


Figure 4.9: Our method is robust to inconsistencies and outliers in the data, such as people walking by (horizontal lines) or plants moving in the wind (jagged green lines; see also plants in Figure 4.1).

standard benchmark datasets. However, please note that most of these algorithms have been designed with different application scenarios in mind. Hence these comparisons are meant to illustrate the novel challenges for the field of image-based reconstruction arising from the ability to capture increasingly dense and higher resolution input images. For each method we hand-optimized parameters and the camera separation of the input images for best reconstruction quality.

Comparison to Two-View Stereo

Comparing the results in Figure 4.10 and focusing on the closeups, issues of existing methods with such highly detailed scenes become obvious. The popular *graph cuts* [Kolmogorov and Zabih, 2001] as well as the more recent *cost volume filtering* approach [Rhemann et al., 2011] are time and memory intensive and could not process resolutions higher than 1 MP. Both methods reconstruct sharp boundaries, but they are not well localized due to the low resolution. Homogeneous image regions are problematic as well. Good performances in terms of memory and



(a) Input | **(b)** Graph cut (~1 day on 1 MP) | **(c)** Cost volume filtering (~1 day on 1 MP) | **(d)** Dense seed-and-grow (10 min) | **(e)** Semi-global matching (4 min on 4 MP) | **(f)** Ours (9 min on 21 MP)

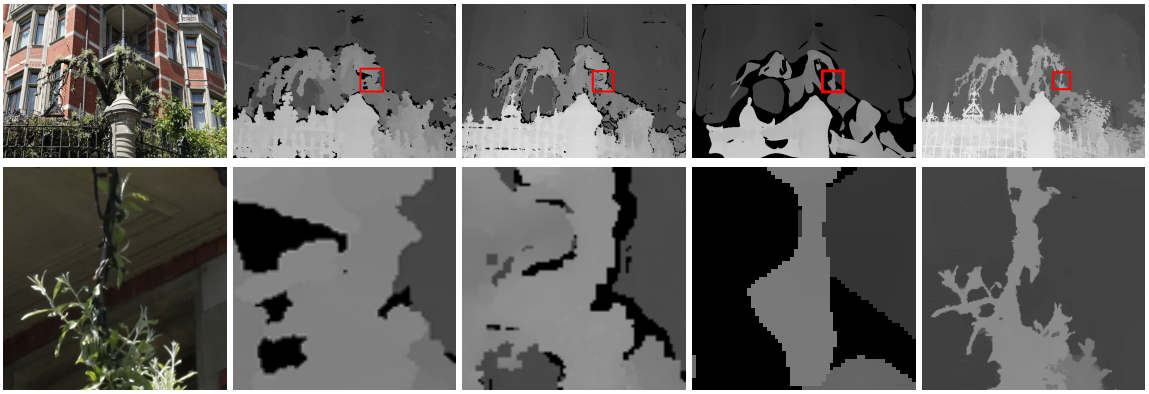
Figure 4.10: Comparison to two-view stereo methods on the Mansion dataset. **(a)** One input image, **(b)** Kolmogorov and Zabih [2001], **(c)** Rhemann et al. [2011], **(d)** Geiger et al. [2010], **(e)** Hirschmüller [2005], and **(f)** ours. The numbers in parentheses denote the running time to compute a depth map, but are measured with different implementations (C/Matlab) and processor types (CPU/GPU).

runtime are achieved by the *dense seed-and-grow* approach of Geiger et al. [2010] and by *semi-global matching* [Hirschmüller, 2005] (as implemented in OpenCV). However, these methods show problems in homogeneous regions and around object contours as well (see black pixels). Leveraging the huge amount of data in a corresponding light field of the scene, our method reconstructs detailed, well-localized silhouettes and plausible depth estimates in homogeneous regions at reasonable run times.

Comparisons to Multi-View Stereo

In Figure 4.11 we show results of recent multi-view stereo methods. For comparison to our result in Figure 4.11e we show a 3D rendering of the point clouds which is colored in accordance to depth and selected a similar closeup region as before. The method of Furukawa and Ponce [2010] leverages information from 50 views of the light field. We also compare against the method of Beeler et al. [2010] that was originally developed for high quality face reconstruction and that uses 8 input images. As it is optimized for faces, its core assumptions regarding smoothness and surface continuity are violated, hence the authors processed our dataset running only the initial multi-view matching part of their pipeline. Overall both approaches achieve good reconstructions, but lack details around contours and miss some homogeneous regions in comparison to our method. We also show a result produced using a commercial tool, *Autodesk 123D Catch*[2] that to our knowledge is based on

[2]http://www.123dapp.com/catch



(a) Input | **(b)** Furukawa and Ponce [2010] (6 min on 10 MP) | **(c)** Beeler et al. [2010] (20 min on 10 MP) | **(d)** 123D Catch (5 min on 10 MP) | **(e)** Ours (50 min on 21 MP)

Figure 4.11: Comparison to multi-view stereo methods. Each method was supplied with as many views as it can process up to 50 views. The numbers in parentheses measure the time required to compute the full reconstruction, e.g., 50 depth maps for our method.

the work of Vu et al. [2009]. The application could process 10 images and produced a very smooth result that, however, lacks any detail.

Comparisons Using Stereo Data Sets

We also ran our method on classic stereo data that has been used in the stereo community for benchmarking. These datasets differ significantly from the fundamental assumptions behind our algorithm as they encompass a relatively small number of low resolution input images. In Figure 4.12 we show our result on the Flower Garden sequence[3] (50 images, 0.08 MP). On this small spatial resolution, our method takes about 3 seconds to compute a depth map with quite accurate silhouettes. However, due to missing texture in the sky, artifacts in the top left corner arise.

In the following we show additional comparisons on classic stereo data sets of Scharstein and Szeliski [2002], widely known as Middlebury data sets, and of Zitnick et al. [2004]. For this low spatio-angular resolution data the quality degrades tangibly as our method has been specifically designed to operate on the pixel level by leveraging highly coherent data. In such scenarios, methods employing comparisons of whole image patches and global regularization are advantageous.

In Figure 4.13 we compare our method to two stereo methods [Zitnick and Kang, 2007; Szeliski and Scharstein, 2002]. Both methods initially match image segments or patches and then refine these coarse estimates using a smoothing or propagation

[3] http://persci.mit.edu/demos/jwang/garden-layer/orig-seq.html



Figure 4.12: Result on the Flower Garden sequence with 50 images. *Left:* One input image with 0.08 MP resolution. *Right:* Our depth map. The computation time was 3 seconds.

strategy. For evaluation we use Middlebury stereo data sets with the ground truth[4], enabling a quantitative comparison. Note that we use all input images (5 images for *Tsukuba* and 8 images for *Venus* and *Sawtooth*) whereas the other methods use two images only. In Table 4.1 we compare the estimation errors, for which we compute the percentage of bad estimates. We consider an estimate bad if its difference from the ground truth disparity is larger than a threshold T.

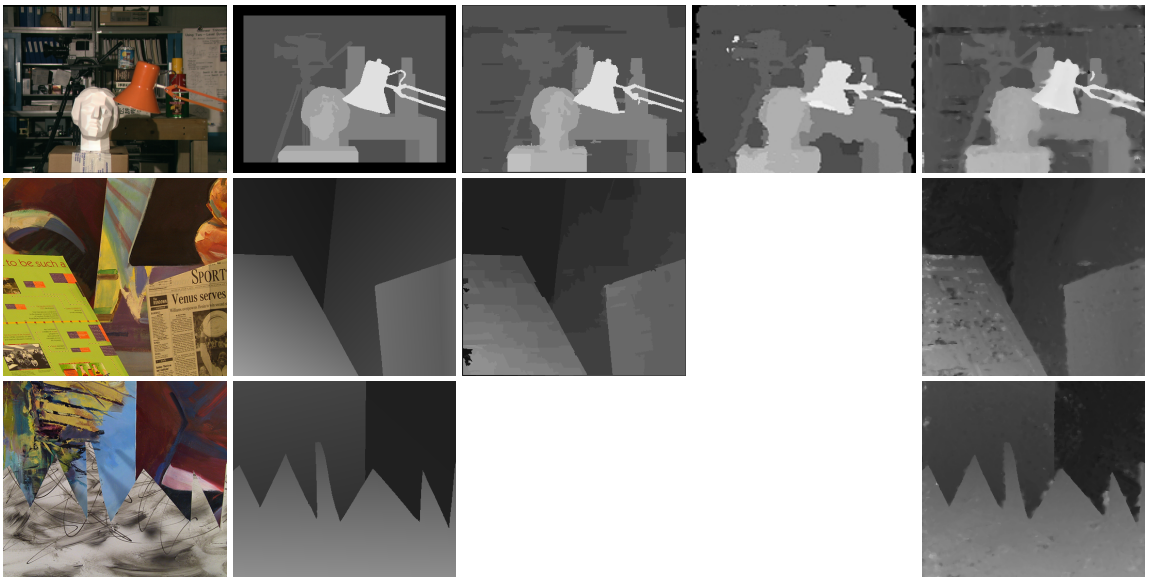
	Tsukuba	Venus	Sawtooth
Zitnick and Kang [2007]	1.87 %	1.85 %	–
Szeliski and Scharstein [2002]	4.9 %	–	–
Ours	8.42 %	10.59 %	6.25 %
Run time	*1.4 s*	*2.4 s*	*2.6 s*

Table 4.1: Quantitative comparison on Middlebury stereo data. We report errors as the percentage of bad pixels with $T = 1$.

It takes about two seconds for our method to process these data sets as they are both angularly and spatially at low resolution (5 or 8 images with 0.1–0.2 MP each). Szeliski and Scharstein [2002] report 4.7 seconds for *Tsukuba* data set, which was measured on a 750 MHz Pentium III CPU. Zitnick and Kang [2007] do not provide run times. The quality of our results for these data sets is not optimal. This can be due to the low spatio-angular resolutions since our method is specifically designed to operate at the pixel level by leveraging the redundancy and coherence in high resolution light fields. For such data sets, methods based on image patch comparisons and global regularization perform better.

In Figure 4.14 we compare our method to the multi-view stereo method of Zitnick et al. [2004] using their data sets[5] consisting of videos captured by eight synchronized cameras at a resolution of 0.8 MP. Although the data sets are at a higher resolution

[4] http://vision.middlebury.edu/stereo/
[5] http://research.microsoft.com/en-us/um/people/larryz/videoviewinterpolation.htm



(a) Input **(b)** Ground truth **(c)** Zitnick and Kang [2007] **(d)** Szeliski and Scharstein [2002] **(e)** Ours (~2 s)

Figure 4.13: Comparison to stereo methods on Middlebury datasets with ground truth. Shown are *Tsukuba*, *Venus* and *Sawtooth* data sets from top to bottom.



(a) Input **(b)** Zitnick et al. [2004] **(c)** Ours (12 s)

Figure 4.14: Comparison to the multi-view stereo method of Zitnick et al. [2004] on their 8-view *Breakdancing* (top) and *Ballet* (bottom) data sets.

than Middlebury stereo data, we found them particularly challenging for our method. The main reasons are considerable noise and exposure changes between cameras. As in the previous comparison, the assumptions our method is based on do not hold with these data sets.

4.4.3 Applications

Scene reconstruction finds a number of immediate uses in applications related to computer graphics besides generating a 3D model of a scene. In the following we illustrate how the output of our method can be directly used for applications such as automatic image segmentation as well as image-based rendering.

Segmentation

Despite being a common task in movie production, automatic segmentation like background removal is still a challenge in detailed scenes. Due to the precise object contours in our reconstructions we can use our method for automatically creating high quality segmentations. For the shown results we simply thresholded all pixels within a prescribed depth interval. Using our depth this approach is not only easy to implement, but also supports real-time updates to the segmentation even on the high resolution images. In Figure 4.15 we show results on the Mansion data set. We wish to stress that such results would be very difficult to obtain using classical color-based or manual segmentation due to the extreme detail in this scene and the partially similar colors between foreground and background.

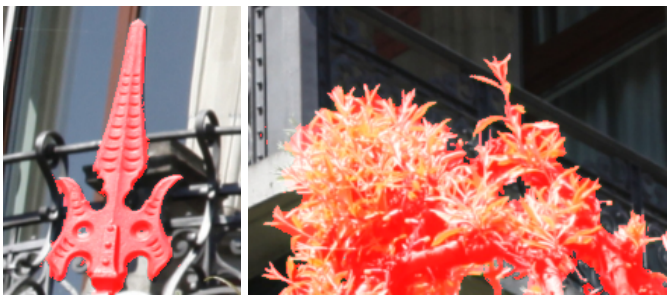


Figure 4.15: Closeups of depth-based segmentations of the Mansion dataset. Note the high level of detail and that foreground and background would be very difficult to distinguish solely based on color.

Image-Based Rendering

Another benefit of our method is that we get consistent depth estimates for any input view of the light field, i.e., we compute as complete a scene reconstruction as

possible from the available input data. Thus, we can directly visualize our results as a colored 3D point cloud using splat-based rendering, with the ability to look around occluding objects (see Figure 4.16). Moreover, we can use the delta EPI representation to reproduce view dependent effects during rendering, e.g., using a weighting scheme as proposed in Buehler et al. [2001].



Figure 4.16: Examples for novel view-synthesis by rendering a colored point cloud. The leftmost image is from the set of input images.

4.4.4 Results for 4D and Unstructured Light Fields

A result for a 4D light field from the Stanford Light Field Repository[6] is shown in Figure 4.17 where we also provide a visual comparison to the 4D light field depth estimation method by Wanner and Goldluecke [2012a]. While they achieve already appealing results, our method resolves additional details, e.g., on the wheels and

[6] http://lightfield.stanford.edu/lfs.html



(a) Image

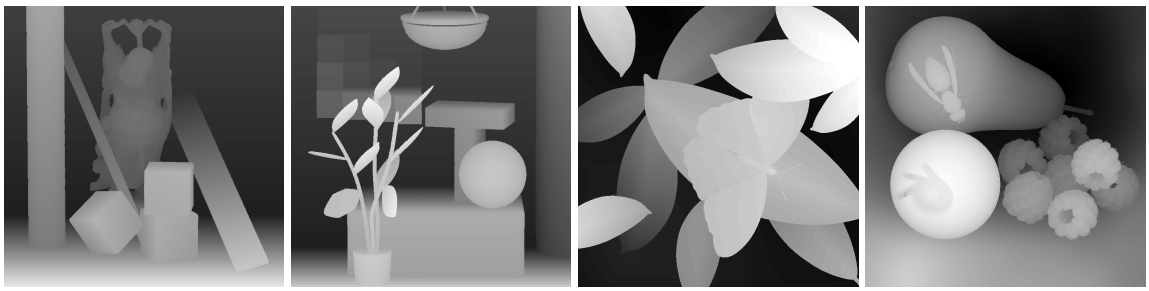
(b) Wanner and Goldluecke [2012a] (15 min)

(c) Ours (1 min)

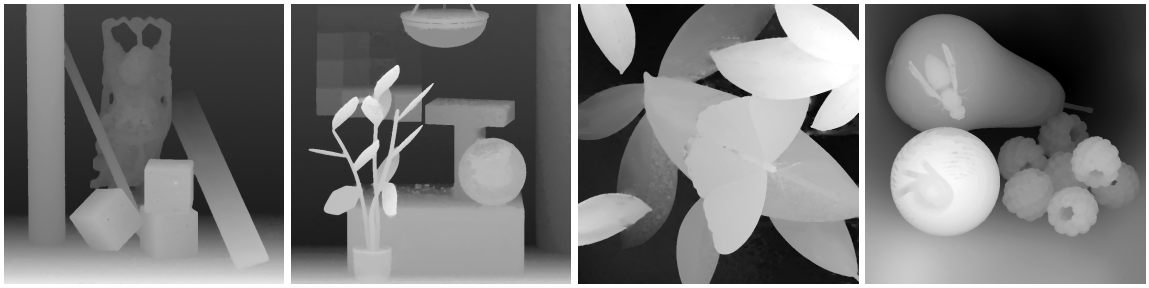
Figure 4.17: Comparison of **(b)** globally consistent labeling of Wanner and Goldluecke [2012a] to **(c)** our result on a 4D light field.



(a) Input



(b) Ground truth



(c) Our results (28 s)

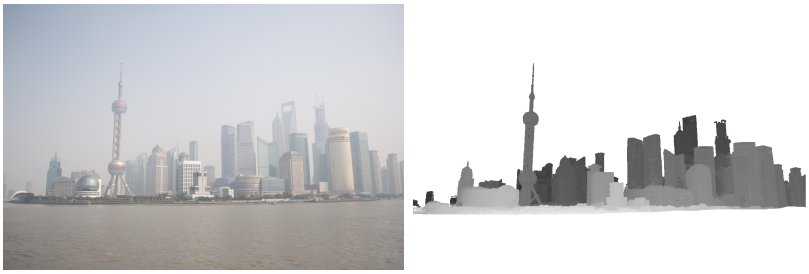
Figure 4.18: Results on 4D HCI light field data with ground truth. Results from *Buddha*, *Mona*, *Papillon* and *StillLife*, from left to right, data sets are shown.

the small holes in the Lego bricks. They report a timing of 15 minutes, whereas ours takes 64 seconds.

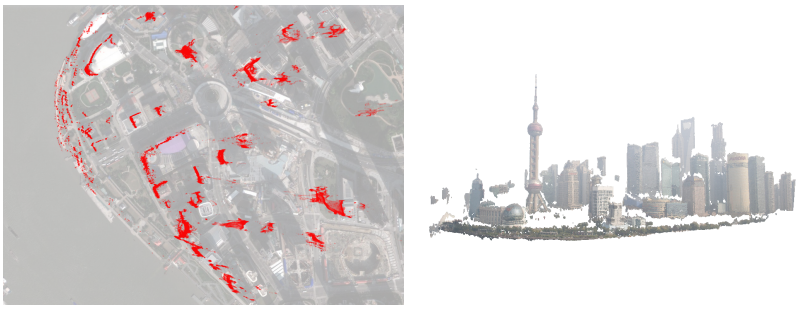
In Figure 4.18 we use synthetic light field data from the HCI lab[7] to quantitatively evaluate our results on 4D light fields using the available ground truth depth. The error measures are summarized in Table 4.2 where we use the same measurement as in Section 4.4.2. We report two error measures with different threshold values to count the bad estimates. As can be observed, our method produces high quality depth estimates on this data. Processing 9×9 images at a resolution of 0.6 MP requires 28 seconds.

In Figure 4.19 we show an example for a challenging hand-held capture scenario. The input images have been taken on a boat in front of the skyline of Shanghai,

[7] http://hci.iwr.uni-heidelberg.de/HCI/Research/LightField



(a) One input image **(b)** Depth map



(c) Map overlay **(d)** Rendering

Figure 4.19: Results on a challenging unstructured light field, obtained by hand-held capture, **(a)**, from a floating boat. **(b)** A resulting depth map. **(c)** Overlay of our reconstruction on a satellite image © 2013 DigitalGlobe and Google. **(d)** Rendering from a novel viewpoint.

with considerable variation in orientation of the camera and of the colors within the scene. We segmented the sky and the water surface. To assess the quality of our reconstruction we also show a bird's eye view overlaid on a satellite image of this area. Computing depth took 162 seconds per view at 3 MP spatial resolution using 100 images. For such unstructured input we observed an increase in running time of about 50% compared to structured 3D input.

	Buddha	Mona	Papillon	StillLife
Bad estimates ($T = 0.1$)	0.89 %	3.81 %	4.93 %	4.33 %
($T = 0.5$)	0.21 %	0.27 %	0.34 %	0.49 %

Table 4.2: Quantitative comparison on 4D light fields. We report errors as the percentage of bad pixels using different threshold values T.

4.5 Discussion

We presented a method for scene reconstruction from densely sampled 3D light fields. A limitation of our method are surfaces with varying directional reflectance, as they violate the assumptions behind the radiance density estimation. This is for example apparent in the reconstruction of the metallic car surface on the bottom left in the Statue dataset in Figure 4.7. This dataset also contains comparably large homogeneous areas in the background, leading to slightly noisy depth estimates in these regions. In some cases, however, like for the windows in the Mansion dataset, the combination of our confidence measures and the fine-to-coarse approach succeeds in plausibly filling even such difficult regions. However, a more principled approach would of course be desirable, e.g., following Criminisi et al. [2005], and it would be worth investigating, e.g., to combine our ray density estimation with more sophisticated reflectance models. Low contrast between foreground and background objects over the whole light field may also lead to problems, as witnessed on some parts of the cables in the Church sequence in Figure 4.7. Finally, while our reconstructions feature precise contours and are very complete as they produce a depth estimate for every input ray, we achieve lower accuracy in terms of absolute distance measurements than a laser scanner. To improve accuracy, investigating a continuous refinement of our discrete depth labels also seems promising.

While the reconstruction of static scenes already has a number of applications, extending our method to temporally varying light fields of dynamic scenes, e.g., using an array of high resolution cameras, provides many interesting new opportunities and challenges. We believe that such very high resolution data may require a rethinking of existing algorithm designs, e.g., using global optimization.

5

Geometry-Driven Sampling Analysis

Geometric information such as depth obtained from light fields finds more applications recently, as shown in Chapter 4. Where and how to sample images to populate a light field is an important problem to maximize the usability of information gathered for depth reconstruction. In this chapter, we formulate this as a view sampling problem. We propose a simple analysis model for view sampling and an adaptive, online sampling algorithm tailored to light field depth reconstruction. Our model is based on the trade-off between visibility and depth resolvability for varying sampling locations, and seeks the optimal locations that best balance the two conflicting criteria.

5.1 Introduction

The attempts to recover geometric information such as depth of the scene captured in the light field is gaining more and more attention. Not only does it play an important role for rendering the light field [Gortler et al., 1996; Buehler et al., 2001], super-resolving it [Bishop et al., 2009; Wanner and Goldluecke, 2012b], or finding the focus plane for post-capture refocusing [Ng et al., 2005; Venkataraman et al., 2013], but it also finds its way in 3D reconstruction and shape acquisition [Wanner and Goldluecke, 2012a; Heber and Pock, 2014]. Understanding the underlying sampling properties is important to maximize the gain from the effort of acquiring the light field. However, the sampling properties of the light field have been mostly studied in the context of reconstructing the original plenoptic function and rendering it from different perspectives by re-sampling process [Chai et al.,

2000; Durand et al., 2005; Levin et al., 2008a; Levin and Durand, 2010]. These are classical sampling reconstruction problems where radiance is measured at a number of locations in the multi-dimensional domain and the plenoptic function is reconstructed from the sampled values. If each ray $\mathbf{m}$ is seen as a point in 5D space $\mathcal{D} \equiv \mathbb{R}^5$, then light field reconstruction amounts to reconstructing the height field $L(\mathbf{m})$ over this 5D domain.[1]

The problem of depth reconstruction, however, relies on the ill-posed step of finding *correspondences* within this set of light rays. The caliber of depth reconstruction depends crucially on the accuracy of this step, which in turn to a great extent relies on where the rays themselves are sampled—we formulate this as a *view sampling* problem. Although a closely related topic of view selection or planning has been studied in the computer vision and robotics communities, often the proposed methods are tightly coupled to a specific reconstruction scheme and do not generalize well to others [Goesele et al., 2007; Gallup et al., 2008] or do not necessarily focus on the specific nature of the currently popular light field acquisition setups [Olague and Mohr, 2002; Hornung et al., 2008; Furukawa et al., 2010]. We try to bridge the gap between the light field sampling analysis that has been done mostly regarding rendering, and the view sampling that lacks the consideration of the light field. We propose a sampling analysis that is tailored to this particular domain.

We exploit two basic observations:

- A large displacement of the camera between view samples potentially confuses algorithms that find correspondences since it is possible that locations previously visible are now occluded by other objects in the scene;
- However, if successive views are "too close" to each other, so that features move by very tiny amounts over image space then it becomes increasingly challenging for correspondence algorithms to resolve the displacement [Szeliski and Scharstein, 2003].

The right choice of displacement further depends on many factors such as the resolution of the image, size of the camera sensor, distance to the scene, the nature and scale of the scene, etc.

Motivated by these two observations we develop a simple but general *sampling analysis model* and an *online sampling algorithm* based on it, to estimate "good" placement of the camera for high fidelity depth reconstruction. For the derivation, we assume that the sampling locations are restricted to a line,[2] i.e., 3D light fields

[1] More precisely, the 5D domain $\mathcal{D} \equiv \mathcal{S}^2 \times \mathbb{R}^3$, where $\mathcal{S}^2$ denotes a 2-sphere, since the 3D geometry of a ray is determined by a position $(x, y, z) \in \mathbb{R}^3$ and a direction as a polar angle $(\theta, \phi) \in \mathcal{S}^2$.

[2] Likewise, the 3D domain can be more precisely defined as $\mathcal{D} \equiv \mathcal{S}^2 \times \mathbb{R}$, where the ray origin is determined by a single parameter along a line.

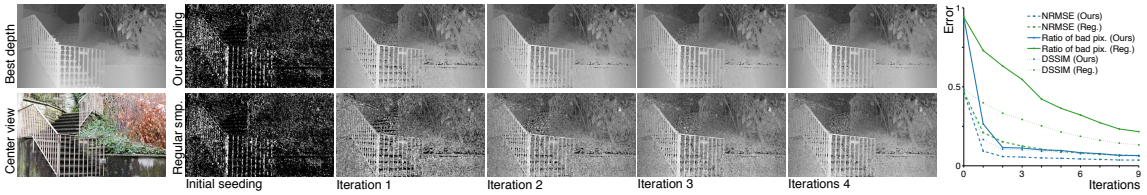


Figure 5.1: Resulting depth maps computed after several iterations with 2 views sampled at each iteration. Our sampling strategy (*top row*) is compared to regular sampling (*bottom row*). The plots on the right display the errors at each iteration against the best possible depth (the lower, the better) and show the faster convergence of ours towards lower errors.

with $\mathcal{D} \equiv \mathbb{R}^3$, and that the analysis is seeded with an inaccurate depth map, which could be obtained using any reconstruction method. Given this, we analyze simple statistics of the scene by trading off problems due to occlusion with those due to depth resolvability, and successively determine locations for camera placement improving the quality of reconstruction most.

Our model considers the very scene being captured and the correspondence algorithm used for reconstruction to gather statistics. Our sampling algorithm uses the model to identify a small set of sampling locations, and successively amasses statistics of the scene, which in turn helps make better view placement in an iterative manner. For example, Figure 5.1 reports the gain in accuracy of selectively locating a small number of views over a few iterations, compared to the regular sampling of the same number of views. While the reconstruction method of Chapter 4 is of our biggest concern, we design our approach as general as to be applicable to any depth reconstruction algorithm, in which a comparable reconstruction quality is achieved with a smaller number of images.

5.2 Sampling Analysis Model

The problem of depth reconstruction relies on determining potential intersections of rays. For this, we define an operator $\mathcal{C} : \mathcal{D} \times \mathcal{D} \rightarrow \{0, 1\}$ that, given two rays $\mathbf{m}_1, \mathbf{m}_2 \in \mathcal{D}$, returns 1 if and only if the two rays originate from the same 3D point. The process of evaluating this operator is known as *correspondence matching* and the implementation of $\mathcal{C}$ has been a long-standing open problem in computer vision. Any algorithm that attempts to be clever with view placement for depth reconstruction must account, in some way, for $\mathcal{C}$.

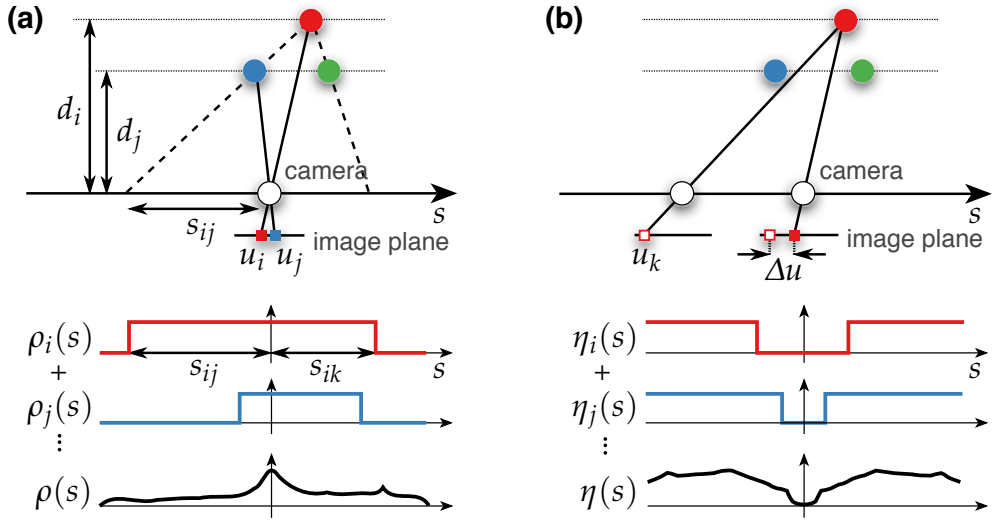


Figure 5.2: Determining sampling intervals. **(a)** $\rho_i(s)$ represents an interval of s where the pixel $\mathbf{u}_i$ is visible and thus matching is feasible. **(b)** $\eta_i(s)$ represents an interval where the depth is resolvable for $\mathbf{u}_i$ up to accuracy function $\mathrm{Acc}(\mathcal{C}_{ik})$. The two intervals are each summed over all pixels to form the distributions $\rho(s)$ of non-occluded pixels and $\eta(s)$ of the pixels with resolvable depth, respectively, over s.

5.2.1 Conservative Sampling Interval

Consider a pair of rectified images of an arbitrarily shaped object containing a repetitive texture. If the texture is periodic, then the task of identifying a unique correspondence between pixels is hopeless. However, for a particular pixel, adding a constraint that the camera separation must be small enough to guarantee no occlusion, robustifies the correspondence detection. Formally, we can represent this constraint for each pixel $\mathbf{u}_i = (u_i, v_i)$ as a visibility preference function ρ over the sampling position s:

$$\rho_i(s) = \begin{cases} 1 & \text{if } \alpha_i \leq s \leq \beta_i \\ 0 & \text{otherwise} \end{cases} . \tag{5.1}$$

Here, $[\alpha_i, \beta_i]$ is the interval along s where the scene point projecting to $\mathbf{u}_i$ is guaranteed *not* to be occluded.

5.2.2 Determining Visible Intervals

Assume that the approximate depth at $\mathbf{u}_i$ is given d_i (see Figure 5.2a). Let s_{ij} denote the distance along the baseline where the scene point projecting to $\mathbf{u}_i$ is occluded

by a scene point that projects to $\mathbf{u}_j$. Then, using basic trigonometry

$$s_{ij} = r_{ij}\frac{d_i d_j}{d_i - d_j}, \tag{5.2}$$

where r_{ij} is the image space distance between the pixels. All distances need to be expressed in the same world units. r_{ij} is related to the pixel disparity by $r_{ij} = (u_j - u_i)/f$ where u_i and u_j are the horizontal pixel coordinates of $\mathbf{u}_i$ and $\mathbf{u}_j$, and f is the focal length in pixels. A conservative visibility condition at $\mathbf{u}_i$ guarantees that at least two samples of the scene point projecting to $\mathbf{u}_i$ are visible (and hence can be exactly matched under the Lambertian surface assumption) if the views are within $[\alpha_i, \beta_i]$, where

$$\begin{aligned} \alpha_i &\equiv \max\{s_{ij}\}, \quad \forall j \mid s_{ij} < 0, \\ \beta_i &\equiv \min\{s_{ij}\}, \quad \forall j \mid s_{ij} > 0. \end{aligned} \tag{5.3}$$

5.2.3 Depth Resolution of Correspondence Algorithms

While a small displacement of the camera along the baseline enjoys the advantage of avoiding occlusion, it introduces the difficulty for the correspondence algorithm to be accurate and reliable. The accuracy of the triangulation of scene points using image features increases with displacement along the baseline [Szeliski and Scharstein, 2003]. We use a simple measure for estimating the depth resolution, which depends on the accuracy of the correspondence algorithm $\mathcal{C}$. Say that the scene point that projects to $\mathbf{u}_i$ in a view project to $\mathbf{u}_k$ after the camera is translated s units along the baseline (see Figure 5.2b). As for visibility, we define a preference function η for depth resolution over s:

$$\eta_i(s) = \begin{cases} 1 & \text{if } \mathrm{Acc}(\mathcal{C}_{ik}) > \epsilon \\ 0 & \text{otherwise} \end{cases}. \tag{5.4}$$

where $\mathrm{Acc}(\mathcal{C}_{ik})$ is the accuracy of the operator for the given pixels and ϵ is some chosen threshold. For the results shown in this chapter, we use the displacement between two images $\mathbf{u}_i$ and $\mathbf{u}_k$ of the scene point for the accuracy function regardless of $\mathcal{C}$, i.e., $\mathrm{Acc}(\mathcal{C}_{ik}) \equiv \Delta u = |u_i - u_k|$, and a minimum required displacement for ϵ. See parameter selection in Section 5.5 for details.

5.2.4 Combining Visibility and Depth Resolution

Clearly depth resolution is better when we use a wide separation distance between views. However, the larger the separation, the more likely that the scene point is

occluded at the new location. We balance between these two factors by multiplying the two, per pixel, which results in a density γ_i over s, a direct measure of view-location preference for $\mathbf{u}_i$. Accumulating this preference over all pixels yields

$$\gamma(s) = \sum_{\mathbf{u}_i \in I} \gamma_i(s) = \sum_{\mathbf{u}_i \in I} \rho_i(s)\ \eta_i(s). \tag{5.5}$$

5.3 Online View Sampling Algorithm

One can use the sampling model developed in the previous section to design a new acquisition device provided a rough estimate of $\gamma(s)$ is known a priori. Otherwise, our model can guide the acquisition process that best suits the particular scene to be scanned and the depth reconstruction method being used. This section details our adaptive view-sampling algorithm and discusses implementation details.

5.3.1 Initial Step

We assume a depth reconstruction method that takes a set of images and produces per-view depth maps, and $K \geq 2$ images that are captured at arbitrary locations s_i, $1 \geq i \geq K$. Using the given depth reconstruction method, we first compute initial K depth maps, each of which we use to estimate local distribution $\gamma^{(i)}(s)$. With the known initial sampling locations s_i, the set of local distributions $\gamma^{(i)}(s)$ is summed up to form a single global distribution:

$$\gamma(s) = \sum_{i}^{K} \gamma^{(i)}(s + s_i). \tag{5.6}$$

In principle, the local maxima of $\gamma(s)$ can serve as the next sampling locations. Instead of being directly used for view sampling, however, they are all put into the queue which prioritizes the candidates for next steps.

5.3.2 Iterations

At each iteration, at most K sampling locations with the highest priority are dequeued. They indicate the locations where the largest number of pixels fulfill both sampling criteria, and thus where new images should be taken. Upon acquisition of the new images, only those new ones are used to estimate $\gamma(s)$. One can expect better estimation of $\gamma(s)$ when including all images, due to the improved depth computation. However, we found this additional depth accuracy does not bring

much advantage in practice, and our book-keeping scheme described shortly handles missing or redundant part of estimated $\gamma(s)$ properly. After obtaining the new distribution covering the new sampling locations, again the local maxima are identified and pushed into the queue. These steps are repeated until either a termination criterion is met or the queue becomes empty.

5.3.3 Termination

At the end of each iteration, a depth map can be computed from all images captured thus far. The algorithm stops when the improvement achieved by the last iteration becomes negligible. If the depth computation is expensive and should be minimized, a more practical criterion is to stop when the target number of sampling locations are achieved. After the last iteration, all the images captured so far are used for the final depth reconstruction.

5.3.4 Priority Queue for Sampling Locations

The priority queue maintains the sampling location candidates as tuples (s, w), i.e., for each s_i, the queue also stores its associated preference, $w_i = \gamma(s_i)$. Whenever a new tuple is being pushed, the queue first checks if the location s has been already seen before by looking up the directory $\chi(s)$: if it is marked so, the tuple is discarded. If at least one new tuple is added, the queue re-arranges its tuples in the descending order of w_i. Then, for each location s_i from the highest to the lowest priority, the queue contracts all tuples (s_j, w_j) within some distance ζ from s_i, forming a new tuple

$$(s^\star, w^\star) = \left(\frac{\sum s_j w_j}{\sum w_j}, \max\{w_j\} \right), \ \forall (s_j, w_j) \ \Big| \ |s_j - s_i| < \zeta. \tag{5.7}$$

When dequeued, the location s of the tuple is marked in the directory $\chi(s)$ which records the sampling locations dequeued so far and thus prevents duplicate sampling locations from being used again. In our implementation this directory is discretized at the resolution of ζ. The dequeued sampling location s is quantized to the nearest meaningful value if required. For instance, when applying our algorithm to the 3D light fields that are already captured with regular sampling we set both ζ and the quantization resolution to the sampling distance between adjacent images.

5.4 Experimental Results

We tested the analysis model and the online algorithm with two reconstruction methods: our own depth reconstruction method presented in Chapter 4 and the method of Furukawa and Ponce [2010], a state-of-the-art multi-view stereo reconstruction method best known as PMVS. While the former directly outputs depth maps, the latter results in point clouds, which we projected into the image plane of the reference view to create depth maps. For the computer-generated dataset, we rendered images on demand using Autodesk Maya, a commercial renderer at the exact sampling locations calculated by our algorithm. The ground truth depth was obtained by an extra depth rendering pass. Two more real-world data sets are used, in addition to those used in Chapter 4. These new light fields are video clips captured at the resolution of 1080 HD, while the camera moves on a linear path at a constant speed (see Section 3.3 for more details). This provides us with 3D light fields at a very high angular resolution, comparable to on-demand rendering of computer-generated datasets in practice, at the cost of the reduced spatial resolution. Since the ground truth is not available for captured light fields, we use as the ground truth the best depth map obtained with all available input images and the reconstruction method being used. In addition, we use the images that are the nearest to the predicted sampling locations.

Figures 5.3 and 5.4 show the resulting depth maps over several iterations with $K = 2$, using our reconstruction method in Chapter 4 and the method of Furukawa and Ponce [2010], respectively. The depth maps are computed using increasing number of views whose locations are incrementally determined by our adaptive sampling algorithm. It is compared against regular sampling, where the same number of consecutive images centered around the reference view are used. As seen, our sampling strategy yields better depth maps for all datasets except the Bikes dataset for which ours performs on par with regular sampling. We discuss this later in this section.

Figure 5.5 shows the errors of the computed depth maps shown in Figures 5.3 and 5.4 against the ground truth (or the best possible depth if not available). We used three error metrics: the normalized root-mean-squared errors (NRMSE); the ratio of *bad pixels* whose estimates are different from the truth greater than 5% tolerance; and the structural dissimilarity (DSSIM) that is derived from the structural similarity (SSIM) [Wang et al., 2004] and defined as $(1 - \text{SSIM})/2$. For all metrics, the lower the better. In all plots *blue curves* represent our sampling strategy, while *green curves* the regular sampling. In the last two rows, the ratios between our sampling strategy and the uniform sampling are shown as *red curves* for two reconstruction methods, where the values greater than one means ours performs better and otherwise the uniform sampling works better.

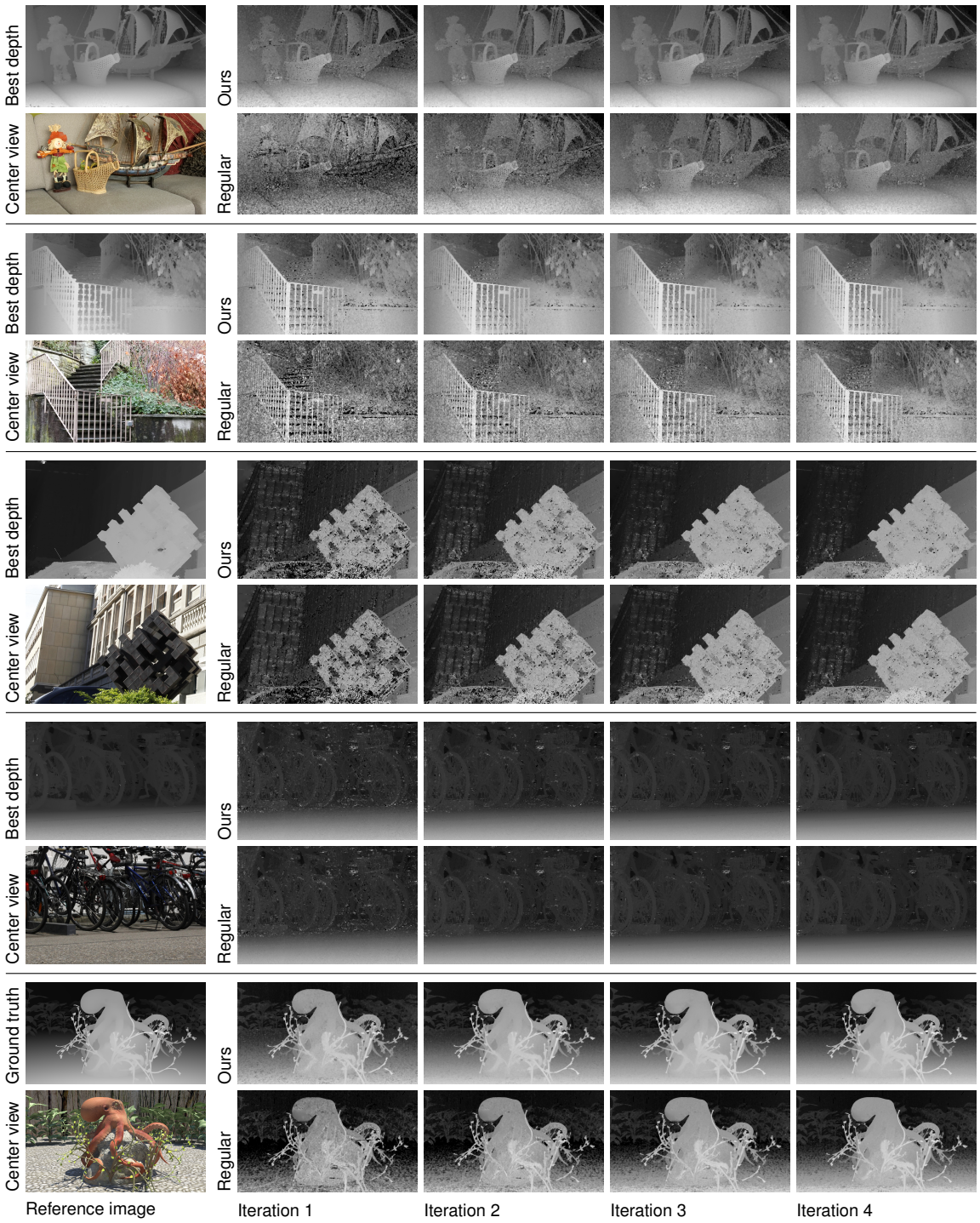


Figure 5.3: Depth maps computed after several iterations with $K=2$, using the reconstruction method presented in Chapter 4. For each dataset, the top row shows the resulting depth maps using our sampling approach, whereas the bottom row shows those using regular sampling. They are seeded with the same $K=2$ sampling positions. A reference image and the best depth map using all views, or the ground true depth for the Octopus dataset, are shown in the first column for each dataset.

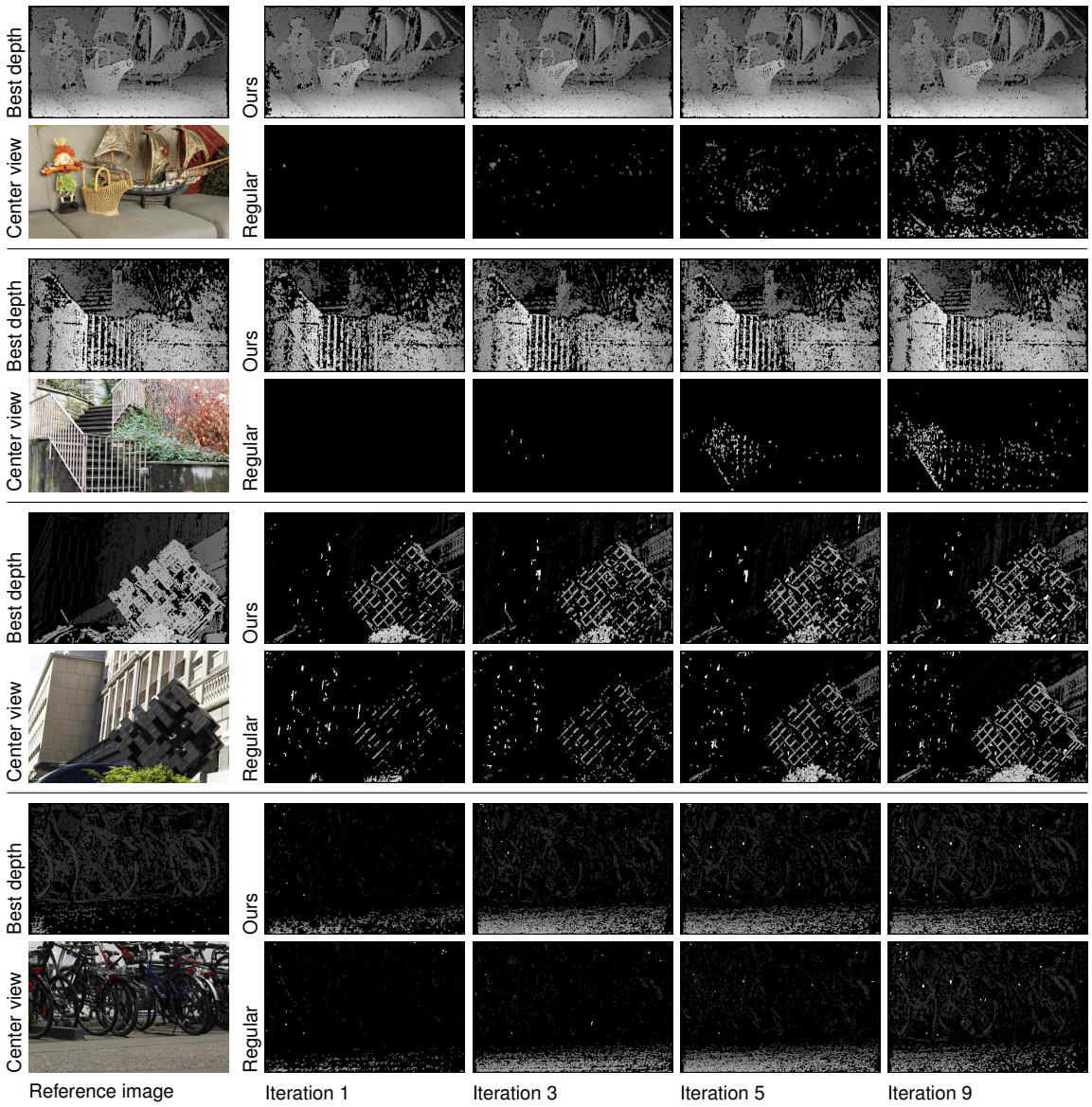


Figure 5.4: Depth maps computed after several iterations with $K = 2$, using the method of Furukawa and Ponce [2010]. As in Figure 5.3, the top row of each dataset shows the resulting depth maps using our sampling approach, whereas the bottom row shows those using regular sampling. They are seeded with the same $K = 2$ sampling positions. A reference image and the best depth map are shown in the first column for each dataset. It took longer to converge, compared to our reconstruction method shown in Figure 5.3, we omitted a few intermediate steps in this figure. Depth maps at the same column were computed using the same number of images.

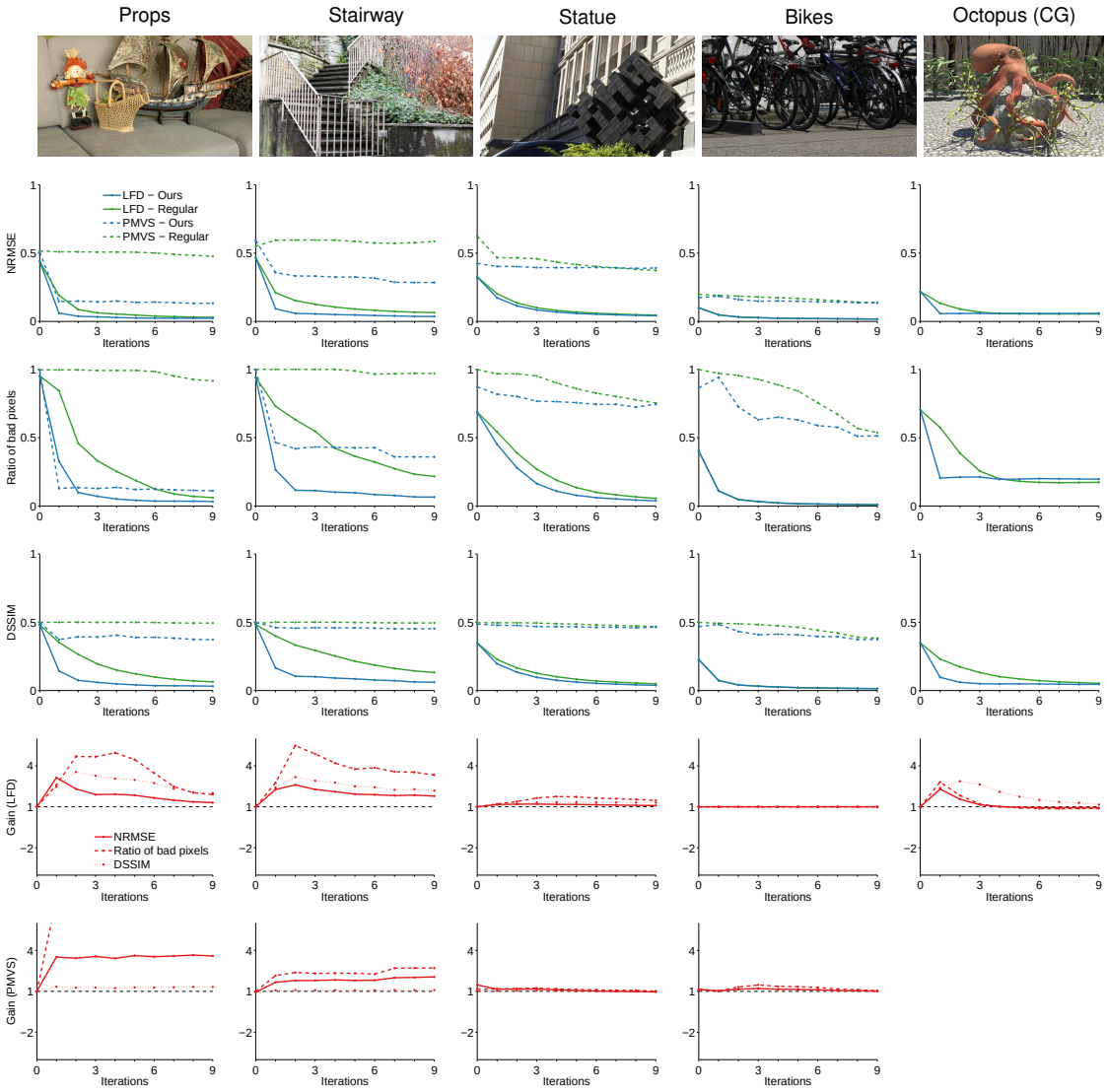


Figure 5.5: Quantitative comparisons between our sampling strategy (*blue curves*) and regular sampling (*green curves*). The two strategies are compared against the ground truth (or the best possible depth if not available) using three error metrics. We used two reconstruction methods: our depth reconstruction method presented in Chapter 4, labeled LFD, and the multi-view stereo reconstruction method of Furukawa and Ponce [2010], labeled PMVS. The first row shows the normalized root-mean-squared error (NRMSE) of the computed depth from the ground truth; the second row the ratio of bad pixels; and the third row the structural dissimilarity (DSSIM) between the computed and the true depth. For all metrics, the lower the better. The red curves at the bottom rows show the gain of our sampling over regular sampling: the plots greater than one mean that ours performs better.

According to our analysis, the Statue dataset is captured at about the optimal sampling rate for depth reconstruction, showing the error curve from regular sampling closely follows the one from our sampling strategy (see Figure 5.5, the third column), whereas the Bikes dataset shows insufficient sampling, where our sampling strategy gracefully degenerates to regular sampling. That is, there are not enough images in between the sampling locations, our sampling algorithm picked the next possible outer images at each iteration. For the other two captured datasets, we have redundant sampling; the sampling locations suggested by our sampling algorithm skips many images at each iteration. The results also show that our sampling algorithm works with a reconstruction method developed with different principles. Although we derived the theory in the context of light fields, the essence of our theory could be extended to other types of reconstruction methods such as multi-view stereo methods.

5.5 Discussion

We proposed a theory for adaptive view sampling motivated by faithful depth estimation from light fields, and presented an online view sampling algorithm based on this theory. Through the experimental validation, we showed that our algorithm converges quicker than regular sampling, and in the worst case, our algorithm degenerates to regular sampling.

Our approach has three properties that will allow for generalization:

- It considers the statistics of the very scene being captured;
- It is not tied to a particular reconstruction algorithm, and uses the algorithm currently used for reconstruction to gather statistics;
- More constraints (preference functions) may be included for view sampling besides occlusion and depth resolution, i.e., more terms in Equation 5.5.

The conceptual extension of our algorithm to view sampling on a 2D plane (e.g., camera arrays instead of a linear stage) should be trivial. All the preference functions will be over a 2D domain and the priority queue will need to compute peaks and distances in 2D instead of 1D. We also believe that this will serve as a first step towards future work on view sampling using unstructured camera locations, i.e., view sampling in 3D.

5.5.1 Parameter Selection

All the results in this chapter were generated using the same, simplistic accuracy function $\mathrm{Acc}(\mathcal{C}_{ik}) = \Delta u$ and the same constant ϵ that is set to be one sensor pixel

size. We deliberately defined them less strictly in the theory to accommodate various types of correspondence matching algorithms. In principle, both $\mathrm{Acc}(\mathcal{C})$ and ϵ must be selected using the appropriate accuracy function of the particular correspondence algorithm. Although we did not tune this parameter in our experiments, we observed consistent results despite the very different natures of the reconstruction methods that we tested with. The second parameter to the algorithm is K, the number of view locations generated at each iteration. We also kept this parameter constant at $K = 2$ for all experiments. We observed, however, that the algorithm converged faster with higher K. In principle, the selection of K depends on the distribution $\gamma(s)$ itself. Selecting an optimal K at each iteration is an open problem for future work.

Since our algorithm is online, it requires an approximate depth map to seed the process of view sampling. Theoretically, using a constant function as the seed depth map is sufficient, i.e., the algorithm can be seen as having a dummy iteration at the start. In practice, we use the depth from the adjacent K images at center as for the regular sampling.

We observed that our algorithm is not sensitive to the spatial resolutions of the intermediate depth map. That is, we may use coarse depth maps from downsampled input light fields (with ϵ scaled appropriately) for the estimation of the $\gamma(s)$ distribution and use the highest resolution depth maps only for the final depth computation. This can provide significant speed-up when the depth reconstruction algorithm turns out to be the bottleneck for performance.

5.5.2 Limitations and Future Work

In general, many reconstruction methods assume Lambertian surfaces and do not properly deal with glossy or specular surfaces. Thus, for such methods, it is desirable to avoid the sampling locations where significant amount of view-dependent effects are observed. To this end, the interval analysis may incorporate the level of inconsistency along the angular axis and guide the sampling locations against problematic areas.

It is observed that the algorithm favors the direction where it can see *new* information behind occlusions. This is because of the greedy nature of the algorithm, and may cause it to be stuck in a local minimum. Although it can be a useful property in some situations, it may not be desirable if a depth map at a particular view point would be important since this drifting may cause degradation of scene elements dominant in that specific view. In such cases, we could introduce some randomness to the priority queue which draws some random movement towards the reference view, avoiding too much asymmetric drifting.

Our current algorithm is based on online computation of depth, which in some cases, one may not assume to be viable possibly due to the time and other constraints. In such cases, it would be useful to have some approximate estimation of the sampling distribution. This might be bootstrapped by other types of information, such as monocular depth cues or annotated images.

Although we exemplified light fields with a linear camera alignment, the theory does not assume, nor is limited to, such configuration. It would be fruitful to extend the algorithm for 2D camera configurations and even more interesting scenarios such as circular or spherical light fields, or large-scale aerial captures. We believe our work will open a future avenue for such research.

6

Rendering

This chapter addresses three-dimensional rendering of light fields. In particular, it focuses on stereoscopic view generation from light fields to deal with stereoscopic displays and multi-view autostereoscopic (automultiscopic) displays. A framework is presented that allows for the generation of stereoscopic image pairs with per-pixel control over disparity, based on multi-perspective imaging from light fields. The proposed framework is novel and useful for stereoscopic image processing and post-production. The stereoscopic images are computed as piecewise continuous cuts through a light field, minimizing an energy reflecting prescribed parameters such as depth budget, maximum disparity gradient, desired stereoscopic baseline, and so on. We provide two formulations to solve this problem. First we start with a discerete formulation using the color and depth information of light fields and minimize the energy function using graph cut optimization. Then we formulate the entire view synthesis process including the depth computation in a single variational energy functional, which is solved using primal-dual optimization. While the discrete formulation provides more flexible and accurate control over disparity, the variational formulation is more efficient and scalable to higher resolution light fields. As demonstrated in our results, this technique can be used for efficient and flexible stereoscopic post-processing, such as reducing excessive disparity while preserving perceived depth, or retargeting already captured scenes to various view settings. Our method generalizes for multiple cuts, which is highly useful for content creation for multi-view autostereoscopic displays.

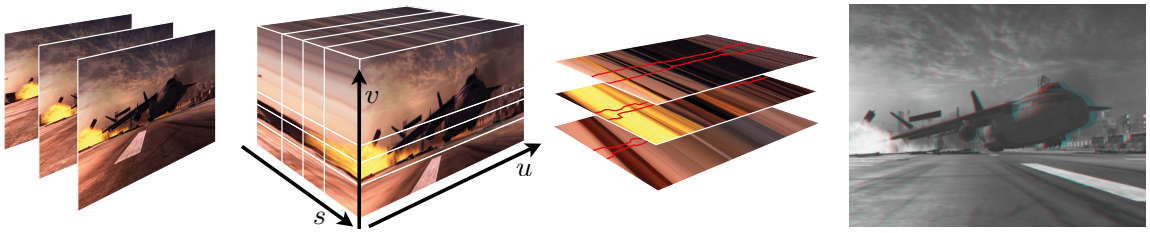


Figure 6.1: We propose a framework for flexible stereoscopic disparity manipulation and content post-production. Our method computes multi-perspective stereoscopic output images from a 3D light field that satisfy arbitrary prescribed disparity constraints. We achieve this by computing piecewise continuous cuts (shown in red) through the light field that enable per-pixel disparity control. In this particular example we employed gradient domain processing to emphasize the depth of the airplane while suppressing disparities in the rest of the scene. Images © 2011 Disney Enterprises, Inc.

6.1 Introduction

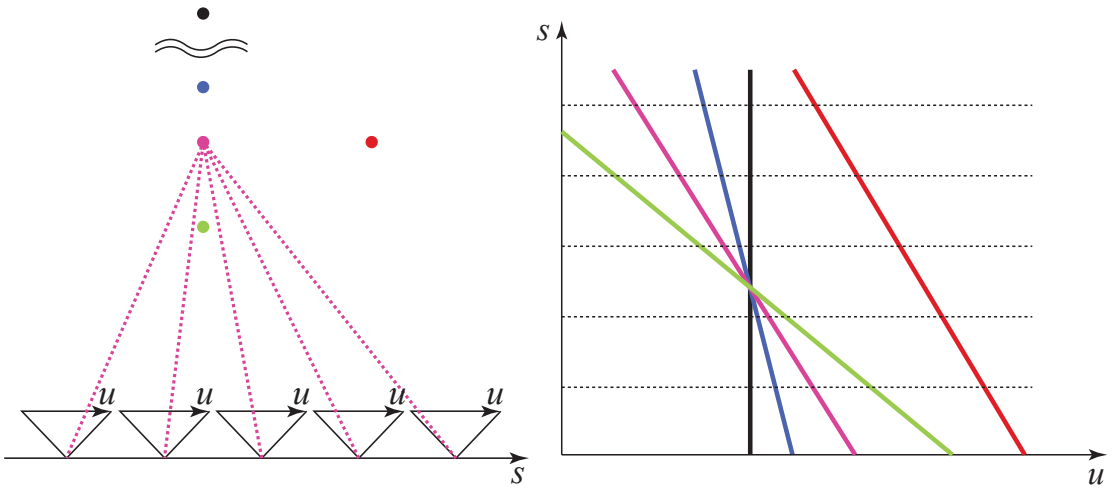
Three-dimensional stereoscopic television, movies, and video games have been gaining more and more popularity both within the entertainment industry and among consumers. An ever increasing amount of content is being created, distribution channels including live-broadcast are being developed, and stereoscopic monitors and TV sets are being sold in all major electronic stores. With new generations of autostereoscopic and multi-view autostereoscopic displays even glasses-free solutions are available to the consumer.

However, the task of creating convincing yet perceptually pleasing stereoscopic content remains difficult. This is mainly because post-processing tools for stereo are still underdeveloped, and one often has to resort to traditional monoscopic tools and workflows, which are generally ill-suited for stereo-specific issues [Mendiburu, 2009]. This situation creates an opportunity to rethink the whole post-processing pipeline for stereoscopic content creation and editing. In the past the computer graphics community has greatly contributed to the development of novel tools for image and video processing. One particular example in the context of this work is the recent progress on light field capture and processing, which enables *post-acquisition* content modification such as depth-of-field, focus, or viewpoint changes. A variety of prototypes for light field acquisition have been developed [Adelson and Wang, 1992; Yang et al., 2002; Ng et al., 2005; Wilburn et al., 2005; Georgiev et al., 2006; Veeraraghavan et al., 2007; Venkataraman et al., 2013] and we already see plenoptic cameras emerging in market such as Lytro. However, the concept of post-acquisition control and editing is missing in stereoscopic post-processing.

The main cue responsible for stereoscopic scene perception is binocular parallax

(or binocular disparity) and therefore tools for its manipulation are extremely important. One of the most common methods for controlling the amount of binocular parallax is based on setting the baseline, or the inter-axial distance, of two cameras prior to acquisition. However, the range of admissible baselines is quite limited since most scenes exhibit more disparity than humans can tolerate when viewing the content on a stereoscopic display. Reducing baseline decreases the amount of binocular disparity; but it also causes scene elements to be overly flat. The second, more sophisticated approach to disparity control requires remapping image disparities (or remapping the depth of scene elements), and then re-synthesizing new images. This approach has considerable disadvantages as well; for content captured with stereoscopic camera rigs, it typically requires accurate disparity computation and hole filling of scene elements that become visible in the re-synthesized views. For computer-generated images, changing the depth of the underlying scene elements is generally not an option, because changing the 3D geometry compromises the scene composition, lighting calculations, visual effects, etc. [Neuman, 2010].

In this chapter we propose a novel concept for stereoscopic post-production to resolve these issues. The main contribution is a framework for creating stereoscopic images, with accurate and flexible control over the resulting image disparities. Our framework is based on the concept of 3D light fields, assembled from a dense set of perspective images. While each perspective image corresponds to a planar cut through a light field, our approach defines each stereoscopic image pair as general cuts through this data structure, i.e., each image is assembled from potentially many perspective images. We show how such multi-perspective cuts can be employed to compute stereoscopic output images that satisfy an arbitrary set of goal disparities. These goal disparities can be defined either automatically by a disparity remapping operator or manually by the user for artistic control and effects. The actual multi-perspective cuts are computed on a light field, using energy minimization to compute each multi-perspective output image. We provide two formulations to solve this problem. This framework is further extended to drive multi-view autostereoscopic (automultiscopic) displays more flexibly and efficiently by computing multiple cuts through a light field. In our results we present a number of different operators including global linear and nonlinear operators, but also local operators based on nonlinear disparity gradient compression. In summary, our proposed concept and formulation provides a novel, general framework that leverages the power and flexibility of light fields for stereoscopic content processing and optimization.



(a) Scene space **(b)** Ray space

Figure 6.2: Light field parameterization. **(a)** A 2D illustration of a scene and an imaging setup to generate a light field. **(b)** The corresponding 2D light field or epipolar-plane image (EPI). Each point in ray space corresponds to a ray in the light field. Scene points seen in multiple images become *EPI lines* in ray space (see Figure 6.1 or 6.3). The slope of each line is proportional to the distance of the corresponding scene point. For points at infinity (black point) the line becomes vertical.

6.2 Goal-Based Stereoscopic View Synthesis

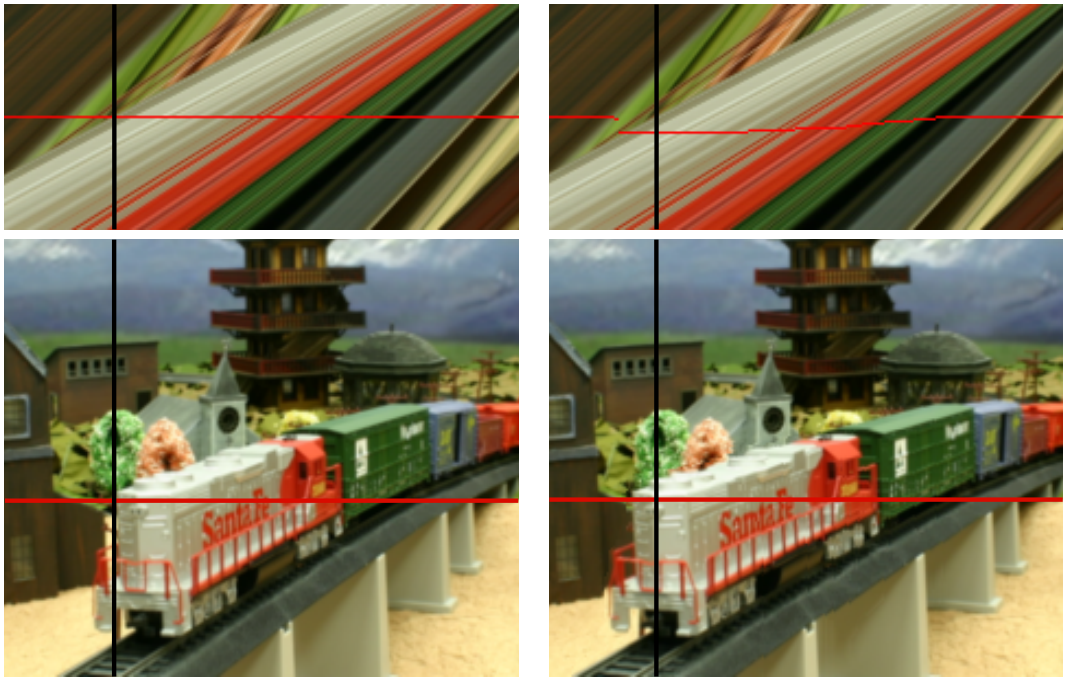
We are interested in generating image pairs for stereoscopic viewing, with accurate control over the corresponding space of binocular disparities, such as range or gradients. More specifically, the images we want to generate should satisfy the stereo constraint [Seitz, 2001], i.e., they should feature *horizontal parallax* only, without any vertical displacement of scene points between the images. Seitz [2001] showed that, in order to satisfy this stereo constraint, the images have to be constructed from a very specific three-parameter family of light rays. This observation is important to the design of our algorithm; instead of having to process full 4D or higher-dimensional light fields, we can focus our discussion on image generation from a 3D light field without loss of generality. In practice, typical examples of setups for 3D light field acquisition are a camera mounted to a linear stage, a linear camera array, or corresponding renderings of a virtual scene. See Figure 6.2a for an illustration of an acquisition setup and Section 3.1.3 and Section 3.3 for a detailed treatment of light field representations and acquisition, which we briefly summarize in the next section.

6.2.1 Image Synthesis from Light Fields

Let $L : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be a 3D light field, created from a set of standard perspective RGB images. Each light ray $L(u, v, s)$ is parameterized by three parameters; parameter s denotes the 1D positional degree of freedom of the ray origin, whereas parameters (u, v) represent the ray direction. Assuming uniform sampling of the ray space with respect to these parameters, Figure 6.2b illustrates a 2D light field corresponding to Figure 6.2a. Figure 6.1 shows an example of an actual 3D light field in the form of an *EPI volume* [Gortler et al., 1996], which can be intuitively interpreted as a stack consisting of the 2D input images. Since the capture process naturally results in a discrete set of rays, the parameters u, v, and s will from now on be implicitly treated as integers. Therefore, s can be regarded as an index to one of the input images, while (u, v) indexes a pixel in image I_s, i.e., $L(u, v, s) = I_s(u, v)$. For simplicity, our discussion will be based on this discretized view of the ray space.

A 2D view, which is not necessarily perspective, can be generated from a 3D light field L by selecting a 2D subset of rays. As a simple example, a planar uv-slice or 2D *cut* at a particular parameter position s extracts the original standard perspective input image I_s (see Figure 6.3a). Cuts with varying parameter s yield images with varying centers of projection. For instance, a vs-cut with constant parameter u results in a so called pushbroom panorama, which corresponds to a sensor with a single pixel column and a linearly varying position of the camera center [Yu et al., 2010]. A us-cut represents a single EPI, i.e., a 2D stack of the same scanline across all images, also illustrated in Figure 6.3. However, there is no restriction to planar cuts. In principle, any 2D subset of rays can be used to generate an image, although a certain ray coherence is required in order to produce "meaningful" images. In the context of stereoscopic image generation, curved, piecewise continuous cuts result in multi-perspective views of a scene, as shown in Figure 6.3b. As shown by Seitz [2001] and Peleg et al. [2001], multi-perspective images can be fused stereoscopically, as long as they feature horizontal parallax only. This observation is the justification for our algorithm that allows the generation of multi-perspective stereoscopic image pairs with controlled disparity by computing corresponding cuts through a light field.

In order to convert a light field cut into an actual image, one has to sample the rays lying on the cut surface. This requires a parameterization of the possibly discontinuous cut which, in general, is a highly difficult problem (related to the field of surface parameterization). However, this problem is further complicated in the context of multiple simultaneous cuts for stereoscopic image generation, since we have to take additional constraints into account. Assume, for example, a straight and a curved cut (as in Figure 6.3) represent a stereoscopic image pair. When sampling the rays along both cuts, any difference in the step size along



(a) Planar cut and the resulting single-perspective image

(b) Nonplanar cut and the resulting multi-perspective image

Figure 6.3: Illustration of a planar, single perspective cut and a nonplanar, multi-perspective cut through an EPI volume. The red line in the bottom images indicates the scanline of the EPI. For easier comparison the black line highlights the same column in both images. Note how the images are nearly identical except for the train front.

the u-axis between the two cuts will have an impact on the horizontal parallax between corresponding scene points in the two images and, of course, also result in different image widths. Similarly, a differing parameterization and sampling along the v-axis will result in vertical parallax, which is undesirable for any stereoscopic image pair. A simple parameterization and sampling strategy, which naturally avoids these issues and does not introduce additional distortion in the output view, is a regular sampling of the cut surface along the u- and v-axis.

The following algorithm combines these basic ideas to compute multiple synchronized cuts through a light field in order to produce multi-perspective images with specific stereoscopic properties.

6.2.2 Stereoscopy from Light Fields

In order to introduce the terms and definitions used for our algorithm, we will first consider the generation of a standard perspective stereoscopic image pair. As discussed in the previous section, one can extract a perspective view from a light

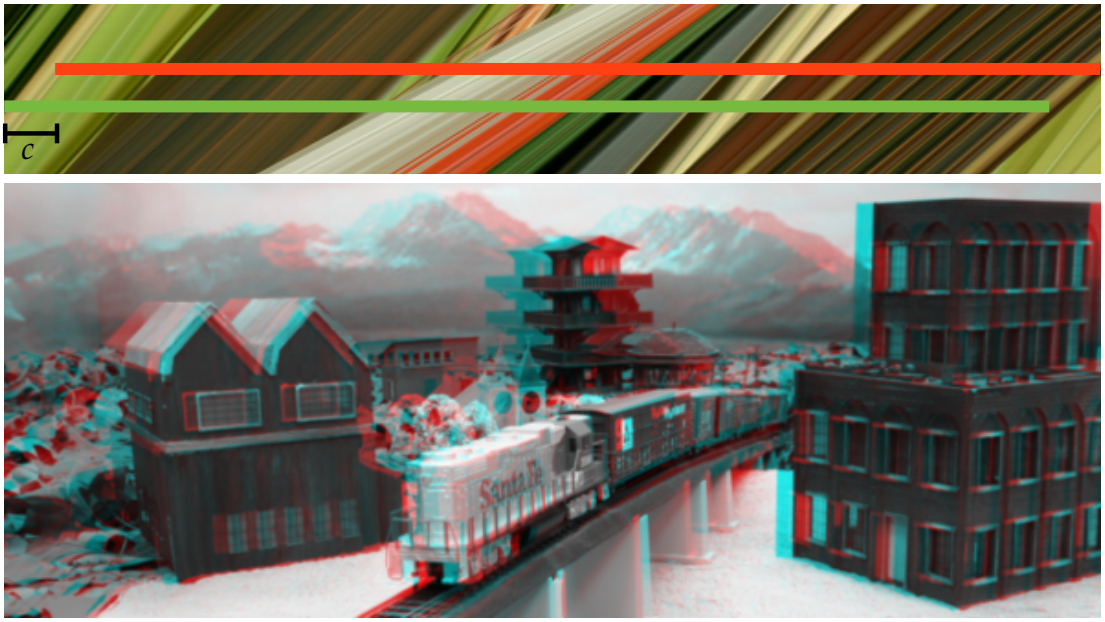


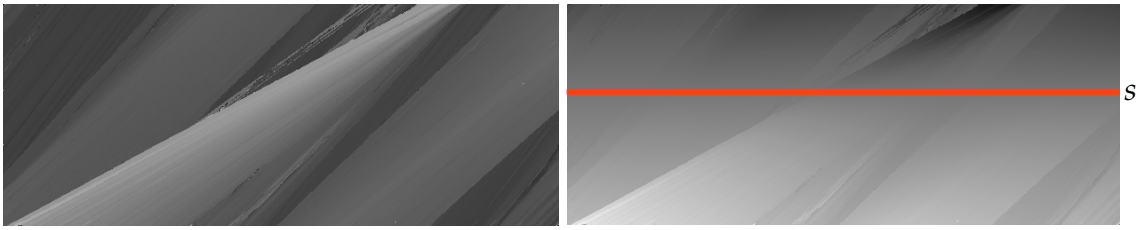
Figure 6.4: A 2D EPI of a light field, showing two planar uv-cuts. The horizontal offset c changes the convergence plane of the stereoscopic image pair. The bottom image shows the corresponding stereoscopic image pair generated from I_s and $I_{s'}$.

field L by fixing parameter s and sampling the rays on the corresponding uv-plane, effectively selecting the input image I_s. As illustrated in Figure 6.2a, different parameters s represent input images captured at different, linearly translated camera positions. Correspondingly, the difference $\Delta(s', s) = s' - s$ is proportional to the camera baseline β between two images $I_{s'}$ and I_s, i.e., $\beta = \Delta(s', s) \cdot b$, where b is the metric distance of unit length in s, or in other words, the distance between adjacent image capture locations. Hence, a stereoscopic image pair with baseline β can be generated by picking a reference view I_s, and selecting the second view at $s' = s + \beta/b$, corresponding to two parallel uv-cuts through L. The convergence c for such a stereoscopic image pair can be modified by shifting $I_{s'}$ horizontally with respect to I_s (see Figure 6.4).

In order to create a stereoscopic image pair from a 3D light field L with constraints on the space of disparities, we define a corresponding 3D disparity volume $D : \mathbb{R}^3 \to \mathbb{R}^+$ that stores the scaled reciprocal of the depth, i.e., the distance measured along the direction perpendicular to the image plane, of the scene point each ray intersects (see Figure 6.5a). D can be interpreted as a *normalized disparity*, such that the image disparity of a pixel $\mathbf{u}$ in $I_{s'}$ to a reference image I_s is defined as

$$T_s(\mathbf{u}, s') = \Delta(s', s) D(\mathbf{u}, s'), \tag{6.1}$$

where we use $\mathbf{u}$ as shorthand notation for the coordinate pair (u, v). While any stereo reconstruction method can be used to construct D from the computed



(a) Normalized disparity D **(b)** True disparity T_s

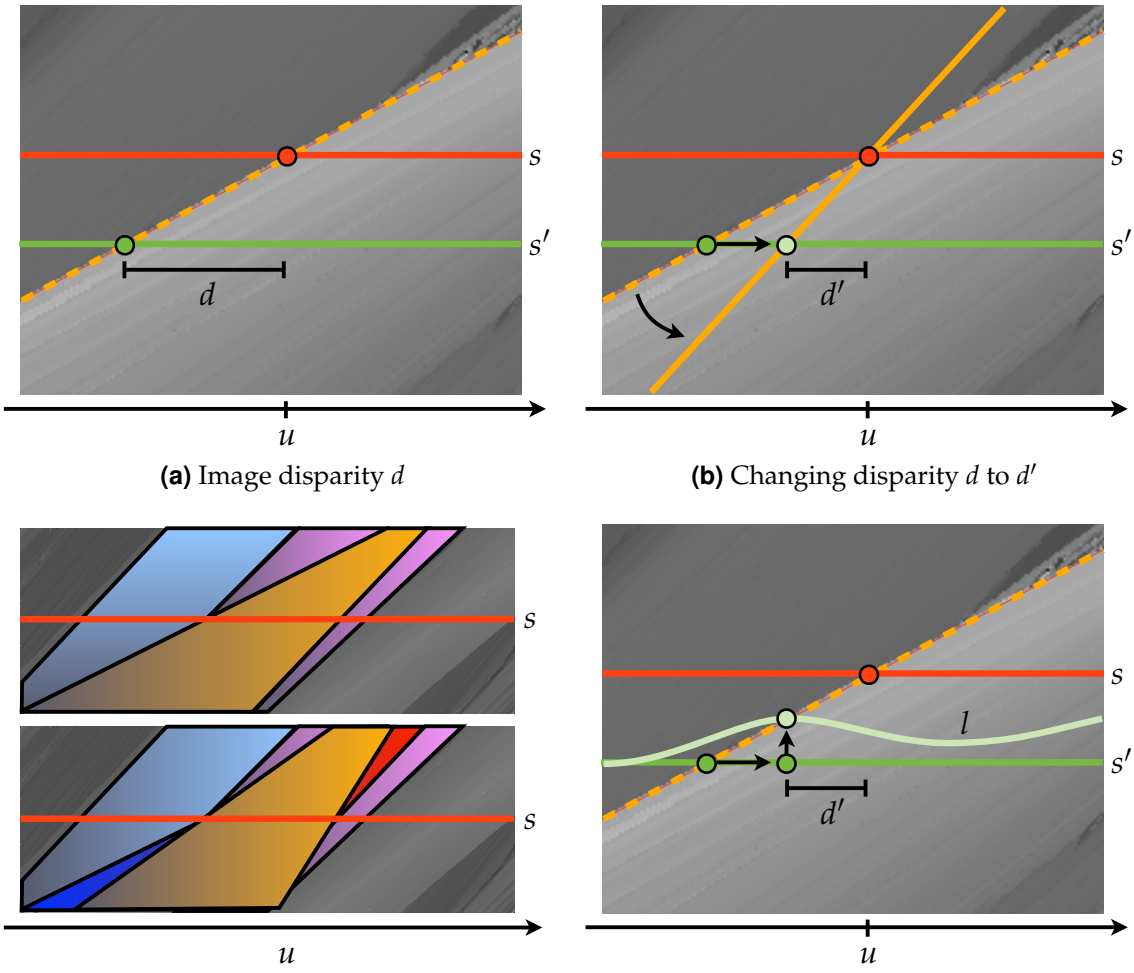
Figure 6.5: (a) A 2D us-slice of the normalized disparity volume D. **(b)** A 2D us-slice of the true image disparity volume T_s with respect to a reference view I_s.

depth maps using Equation 6.1, our reconstruction method in Chapter 4 naturally generates the normalized disparity volume D from the input light field L. For computer-generated scenes, accurate depth can be read from the z-buffer. We call T_s the *true* disparity volume for a particular view I_s, as illustrated in Figure 6.5b.

Given a reference view I_s and the true disparities T_s it is straightforward to formulate a simple procedure that finds a second view $I_{s'}$ such that $T_s(*, *, s')$ does not exceed a certain disparity range. However, the only means for controlling disparity is the distance $\Delta(s', s)$ between the planar cuts. In the following we will describe how to compute nonplanar, multi-perspective views that satisfy more general, content-dependent disparity constraints.

Consider Figure 6.6a, showing a normalized disparity volume D and planar cuts for two images I_s and $I_{s'}$. According to Equation 6.1 the horizontal parallax or image space disparity d of a pixel in I_s to the corresponding pixel in $I_{s'}$ can be computed as $d = T_s(\mathbf{u}', s') = \Delta(s, s')D(\mathbf{u}, s)$. Now assume we want to create a modified stereoscopic image pair that features a different depth impression only for the particular scene point seen at $I_s(\mathbf{u})$. As argued in the previous section, changing $\Delta(s, s')$ globally does not allow for such a local change. An alternative solution is to keep s and s' fixed, and update the actual scene depth $D(\mathbf{u}, s)$ instead by deforming the actual geometry of the scene. The problem with this approach is that modifying the depth of a scene implies changes to the complete underlying light field, since changing the depth of a scene point influences the slope of the corresponding line in ray space (see Figure 6.2 and Figure 6.6b). An example for the consequences is illustrated in Figure 6.6c: reducing the disparity of the frontmost, orange region results in missing light rays in regions further in the back of the scene (depicted in red and blue). The corresponding rays have not been captured in the original light field. Completing those regions would require complex resampling and hole-filling operations on the light field.

Instead of modifying the image distance $\Delta(s, s')$ or the scene depth D, our algorithm computes a nonplanar cut $l : \mathbb{R}^2 \to \mathbb{R}$ through the light field, which maps rays $\mathbf{u}$ to parameters s in order to meet a given set of goal disparity constraints. This idea is



(a) Image disparity d

(b) Changing disparity d to d'

(c) Missing light rays

(d) Multi-perspective solution

Figure 6.6: Multi-perspective light field cuts for changing stereoscopic disparity. **(a)** Given two images I_s and $I_{s'}$ with image disparity d at pixel u. **(b)** Modification of the disparity d to d' effectively amounts to changing the scene depth (see also Figure 6.2), and, hence, the slope of the corresponding lines in the EPI volume. **(c)** Changing depth, in this example of the orange region, results in different (dis-)occlusion patterns, with missing information in the light field (red and blue region). **(d)** We propose to compute a cut l instead, whose corresponding multi-perspective image I_l effectively results in the same change of disparity from d to d'.

illustrated in Figure 6.6d: given the reference image I_s, a second view satisfying the disparity constraint for pixel $\mathbf{u}$ can be generated from a cut l that intersects the EPI line corresponding to $I_s(\mathbf{u})$ at parameter position $u + d'$. Intuitively, the cut l picks for each pixel $I_s(\mathbf{u})$ a pixel from *some* input image, such that the desired disparity constraints are fulfilled. As each input image shows a different perspective of the scene, the cut produces a multi-perspective output image I_l that, together with the reference view I_s, forms a stereoscopic image pair where we effectively control the camera baseline for each pixel individually.

We define the set of goal disparities as a 2D map $G^* : \mathbb{R}^2 \to \mathbb{R}$ that, for each pixel of the *output* view I_l, defines the desired disparity with respect to the reference view I_s as follows. Assume that the disparity of pixel $\mathbf{u}$ in I_s to the multi-perspective image I_l should be d', as shown in Figure 6.6d. This implies that the value of the goal disparity map at position $(u + d', v)$ has to be set to $G^*(u + d', v) = d'$. More generally speaking, let $\varphi : \mathbb{R} \to \mathbb{R}$ be a disparity mapping function that defines how to map the normalized disparity d to a new disparity value. In order to create a corresponding stereoscopic image pair, the goal disparity map then is defined as

$$G^*(u + \varphi(D(u, v, s)), v) = \varphi(D(u, v, s)). \tag{6.2}$$

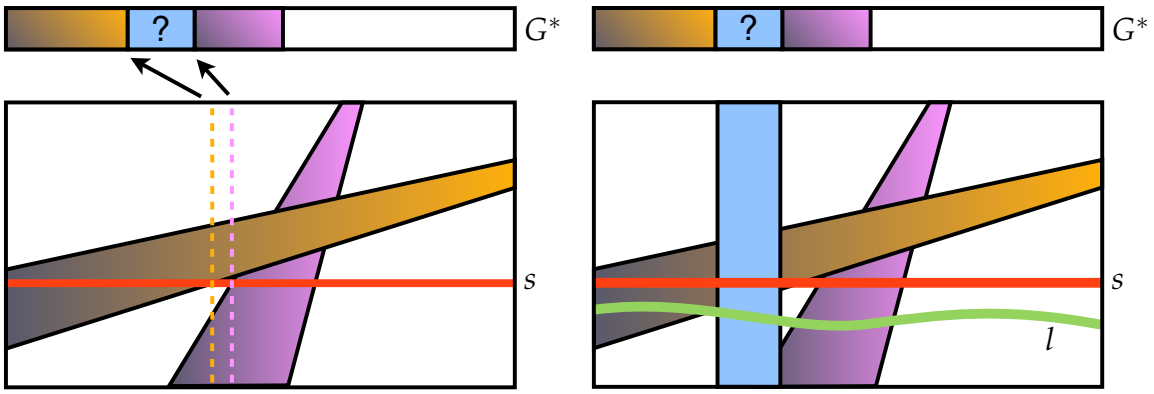
G^* can be constructed by iterating over all pixels $\mathbf{u}$ in the reference image I_s. The asterisk in G^* indicates that it is defined in the *output* view's space. The construction of G^* is neither surjective nor injective due to occlusions and disocclusions in the scene. Intuitively, one cannot define disparity constraints for scene elements that are not visible in I_s. Hence these regions remain undefined in G^* (see Figure 6.7). However, in practice, these monocular regions span only a small number of pixels, hence we can compute a plausible output view by imposing certain smoothness criteria on l, which are described in the following section.

Now recall that the true disparity volume $T_s(u, v, s')$ represents the actual disparity of a point (u, v, s') with respect to I_s; correspondingly, the difference volume $T_s(u, v, s') - G^*(u, v)$ then represents the deviation of a pixel's disparity from the desired goal disparity. The underlying idea of our algorithm for generating the output image I_l is to find a cut l that passes close to the zero set of this difference volume (see Figure 6.8). The following sections describe how the problem of finding l can be formulated as an energy minimization problem.

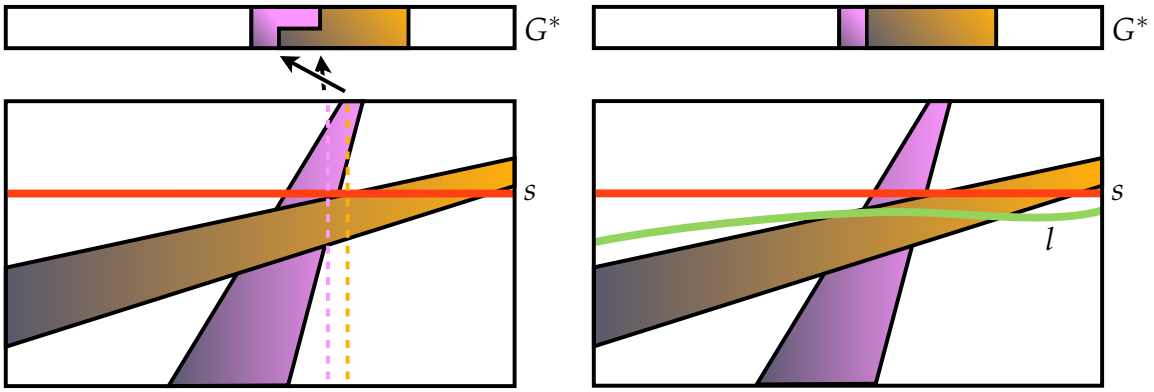
6.2.3 Formulation as Energy Minimization

With the discretization of the light field described in Section 6.2.1, the energy measuring the deviation of a 2D cut l can be expressed as

$$E_{\mathrm{d}}(l) = \sum_{\mathbf{u}} \left| T_s(\mathbf{u}, l(\mathbf{u})) - G^*(\mathbf{u}) \right|. \tag{6.3}$$



(a) Effects of disocclusion and undefined goal disparity



(b) Effects of occlusions and competing goal disparities

Figure 6.7: Illustration of the effects of disocclusions and occlusions. **(a)** Since only depth of scene elements visible in the reference image I_s is known, the construction of G^* by forward mapping of disparities $\varphi(D(*,*,s))$ (see Equation 6.2) is not surjective. This can lead to undefined segments in G^*, illustrated in blue on the left. Intuitively, disparity constraints cannot be defined for regions that are occluded in I_s, but visible in an output view I_l. Since these regions generally span only a small number of pixels, a reasonable choice is to impose a smoothness prior on the cut l. This ensures that the output image shows an undistorted, standard perspective view of all undefined areas, illustrated in blue on the right. **(b)** Similarly, due to visible regions in I_s that will be occluded in other views, the construction of G^* is not injective. Differently remapped disparities of close and distant objects compete for the same range in G^* (overlapping orange and pink region). In this case, we store the disparity constraints for the object closer to the camera (right).

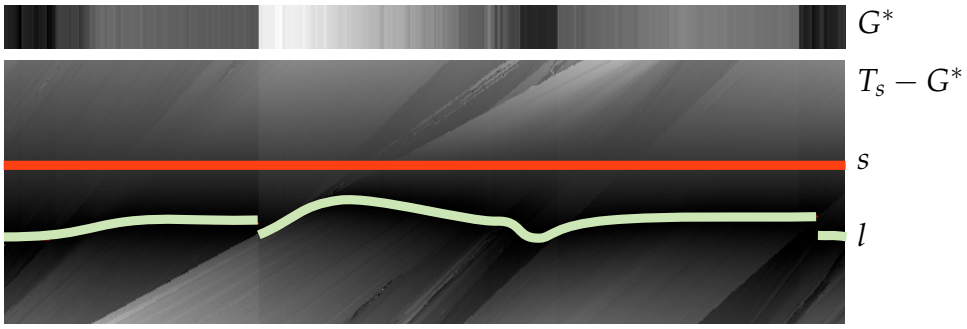


Figure 6.8: Goal disparity. The upper image shows a 1D slice of the 2D goal disparity map G^*. The difference volume $T_s - G^*$, shown as an unsigned function in this figure, then represents the deviation of each point in the light field from the desired disparity. Our algorithm computes a cut l that passes close to the zero set of this volume. The resulting image I_l and I_s then form a multi-perspective stereoscopic image pair with the desired goal disparities.

While a cut computed from this data term alone closely follows the prescribed goal disparities, it does not enforce any coherence between neighboring output rays/pixels and therefore can lead to visual artifacts in noisy or ambiguous estimates of T_s. These artifacts are particularly noticeable in highly textured regions or at depth discontinuities.

Therefore we design an additional content-adaptive smoothness term according to the following observations:

- In the proximity of visually salient parts of an image, such as depth discontinuities and highly textured regions, we would like to enforce higher smoothness to increase the coherence of the rays selected by l. In particular, we would like to assign higher saliency to scene elements close to the camera and cut through more distant regions.
- In visually less salient, homogeneous and continuous regions, smoothness constraints can be relaxed in order to increase the flexibility of the cut to perform multi-perspective view transitions in the light field.

These properties are formulated in the following energy for measuring the smoothness of a cut l:

$$\begin{aligned} E_\mathrm{s}(l) = &\sum_{(\mathbf{u},\mathbf{u}')\in\mathcal{N}_u} \left|l(\mathbf{u}) - l(\mathbf{u}')\right| p_u(*) + \\ &\sum_{(\mathbf{u},\mathbf{u}')\in\mathcal{N}_v} \left|l(\mathbf{u}) - l(\mathbf{u}')\right| p_v(*), \text{ with} \qquad (6.4) \\ p_u(*) = &\min\left\{ p_\mathrm{max}, \left|\partial_s D(*)\right| + \lambda D(*) + \kappa \left|\partial_s L(*)\right| \right\}, \text{ and} \\ p_v(*) = &\min\left\{ p_\mathrm{max}, \left|\partial_s D(*)\right| + \lambda D(*) + \kappa \left|\partial_u L(*)\right| \right\}, \end{aligned}$$

where $\mathcal{N}_u$ and $\mathcal{N}_v$ are the sets of all neighboring pixels along the u-axis and v-axis, respectively. $(*)$ stands for $(\mathbf{u}, l(\mathbf{u}))$. The term $|l(\mathbf{u}) - l(\mathbf{u}')|$ penalizes variation of the cut l along the s-axis, i.e., view transitions. This penalty is weighted by the content-adaptive terms $p_u(*)$ and $p_v(*)$, respectively.

For both axes, the weighted terms depend on the depth discontinuities $\partial_s D$ and the absolute normalized disparity D. Intuitively, for scene elements very close to the viewer, even view transitions to an adjacent view may introduce noticeable disparity jumps. Increasing smoothness for nearby regions and strong depth discontinuities effectively moves view transitions to the background. Note that these concepts can be easily generalized to other types of image saliency, for example to encourage view transitions in less salient regions.

These depth-based terms are sufficient for controlling smoothness of the cut. Optionally, for the u-axis, we can take the change of radiance between different input images I_s into account, while for v we penalize jumps of the cut in the proximity of vertical image edges. Finally, the maximum penalty $p_{\max}$ ensures that the cut can be discontinuous, similar to the concept of robust nonlinear error functions. In our discrete setting, the partial derivatives are computed via forward differences. The constants in Equation 6.4 are only necessary for bringing all terms to a similar scale, but not critical to the quality of the results. For the results in this section we used $\lambda = 0.5, \kappa = 1$, and $p_{\max} = 3$. The final energy is then defined as

$$E(l) = E_{\mathrm{d}}(l) + k\, E_{\mathrm{s}}(l), \tag{6.5}$$

with $k = 25$. One additional interpretation of the smoothness term is that an increased value of k leads to "flattened" cuts, i.e., output images closer to a standard perspective image. We believe that this is a notable property, since higher smoothness does not compromise image quality, but simply falls back to the original input images.

6.2.4 Optimization via Graph Cuts

The minimization of Equation 6.5 can be solved using graph cut optimization [Boykov et al., 2001; Boykov and Kolmogorov, 2004]. We employ the standard procedure for binary s-t-cuts.

- For n input images of dimension $w \times h$ we construct a 3D regular graph of size $w \times h \times (n+1)$.
- A ray at position (u, v, s') is associated with a directional graph edge from node (u, v, s') to node $(u, v, s'+1)$ along the s-axis, and the edge weight is chosen as $|T_s(u, v, s') - G^*(u, v)|$.

- Bi-directional edges between neighboring nodes along the u-axis and v-axis are weighted with the corresponding smoothness values kp_u and kp_v, respectively.
- Boundary nodes with $s = 0$ and $s = n$ are connected to the source and sink of the graph, respectively, with infinite weights.

The min-cut of this graph then yields the desired cut surface l that minimizes Equation 6.5.

We explored various conceptual modifications of this algorithm and the energies. Most notably, we also experimented with additional penalty edges for enforcing C^0 continuity [Rubinstein et al., 2008]. However, we found that piecewise continuous cuts provide more flexibility due to the support for sudden view transitions. Other algorithms for minimizing this energy would be applicable as well. An alternative formulation could be based on multi-labeling via α-expansion [Boykov et al., 2001], where each label is associated with a particular uv-slice along the s-axis of the EPI volume. While such an approach reduces the size of the graph, it has certain restrictions regarding the optimality of the result. In practice, however, we found the binary s-t-cut to produce reliable results.

6.2.5 Extensions

There exist several useful extensions of our basic algorithm which we briefly describe next.

N-View Stereo from Multiple Cuts

Instead of creating a stereoscopic pair consisting of a standard perspective image I_s and a multi-perspective image I_l, the algorithm can be easily extended to create two multi-perspective cuts. For example, two goal disparity maps G^*_{L} and G^*_{R} can be defined as

$$\begin{aligned} G^*_{\mathrm{L}}(u - \frac{1}{2}\varphi(D(u,v,s)),v) &= -\frac{1}{2}\varphi(D(u,v,s)) \text{ and} \\ G^*_{\mathrm{R}}(u + \frac{1}{2}\varphi(D(u,v,s)),v) &= \frac{1}{2}\varphi(D(u,v,s)), \end{aligned} \tag{6.6}$$

where the goal disparities are evenly distributed to both views and the reference view is centered between the two corresponding cuts. More than two views can be handled in a similar manner. As we discuss in Figure 6.14, this multi-cut approach is particularly interesting for content generation for multi-view autostereoscopic displays.

While defining a goal disparity map for each view separately provides high flexibility, many application scenarios such as multi-view autostereoscopic displays often require a simple linear change of disparity between views. This can be exploited for an efficient, interpolation based algorithm to generate multiple views, given just the reference view s and one multi-perspective cut l. Suppose l has been computed from a mapping function $\varphi(D(u,v,s))$, and that the two views s and l should be converted into n views with linearly interpolated disparities. From Equation 6.2 we can conclude that the goal disparities of view $i \in \{0,1,\ldots,n-1\}$ are given as

$$G^*(u + \frac{i}{n-1}\,\varphi(D(u,v,s)),v) = \frac{i}{n-1}\,\varphi(D(u,v,s)), \tag{6.7}$$

meaning that a cut l^i will contain the interpolated points of all EPI lines connecting corresponding points of s and l.

Stereoscopic Video Processing

In order to process video it is generally advisable to enforce a certain continuity between two cuts at consecutive time steps. One solution would be to enforce temporal smoothness by adding a temporal dimension to the graph structure. Each time step then has its own 3D subgraph, and corresponding nodes of subgraphs from consecutive time steps are connected via additional edges. Using a multi-label approach instead of binary labeling, the graph dimension could be reduced to 3D again. The disadvantage of this approach is that it has to process the whole 4D spatio-temporal light field volume at once.

Our solution uses an exponentially decaying influence of previous time steps on the data and smoothness terms for the current time step. Let e_t denote the edge weight for a given time step t according to Equations 6.3 and 6.4. During the update of the graph structure from time $t-1$ to t, we set the temporally averaged edge weight

$$e'_t = \alpha e_t + (1-\alpha)e_{t-1} \tag{6.8}$$

for any edge. However, the temporal evolution of a light field is quite coherent in general. We found a weight of $\alpha = 0.9$ resulting in sufficiently smooth output.

Deferred Rendering for Computer-Generated Content

Our method is particularly interesting for computer-generated content such as 3D animation movies. Implementing multi-perspective camera models into the CG rendering pipeline to meet the expectations of a director regarding control and flexibility is often a difficult problem [Neuman, 2010]. Warping the 3D geometry instead is not an alternative, since this does not allow for arbitrary complex

disparity constraints without compromising the scene composition, lighting calculations, or visual effects. Our method shifts the effort from the artist towards automatic computations: the well-established CG pipeline for modeling, shading, and cameras remains unaltered, and stereography becomes a simple post-process.

However, given the significant rendering time, the generation of the complete light field of a complex scene is not often feasible. To deal with this, deferred rendering could be applied; since our algorithm works well with depth data only (the normalized disparity volume D), it is sufficient to render only the depth maps of the input views. This is typically several orders of magnitude faster than rendering fully shaded color images. Even lower resolution proxy geometry could be used instead of the highly tessellated subdivision surfaces often used in rendering. Once the required set of input views is known from the cut l, those images or just the required light rays can be rendered and combined. These considerations render our method a highly practical solution.

Different Light Field Parameterizations

Section 6.2.1 made certain assumptions about the acquisition and parameterization of a light field, and the required sampling scheme to generate an image from a given cut l. We also assumed that the reference view is a standard perspective view, and that correspondingly our desired output view should be as-perspective-as-possible as well, while satisfying our prescribed goal disparity constraints. For this scenario we argued that a regular sampling along the u- and v-dimension is the most natural choice. In other application scenarios, however, it could be desirable to produce other forms of stereoscopic images, such as omnistereo panoramas as discussed by Peleg et al. [2001], or stereo pushbroom panoramas and cyclographs as discussed by Seitz [2001]. For these types of images the light field parameterization and image cut have to be reconsidered.

As mentioned in Section 6.2.1, a stereo pushbroom panorama simply corresponds to a vs-cut instead of a uv-cut. This insight renders handling of stereoscopic pushbroom images straightforward; one has to swap the dimensions u and s in our formulation, and then apply the algorithm as is. For omnistereo panoramas and cyclographs, the 3D light fields are constructed with a rotating camera at a certain offset orthogonal to the rotation axis, yielding a u-v-α volume. Both above mentioned works show that planar v-α slices can be used to produce stereoscopic panoramas. Peleg et al. [2001] also show an algorithm for adaptively changing the camera baseline for each image column. Our concept of multi-perspective, piecewise smooth cuts with a global optimization scheme generalizes these ideas to *per-pixel* control over the baseline (see Figure 6.9 for a comparison).

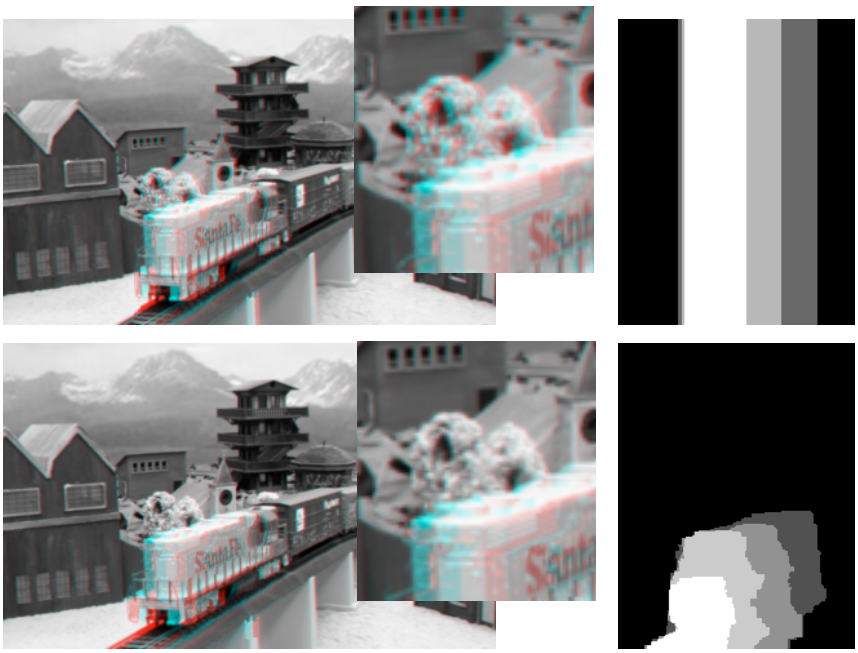
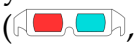
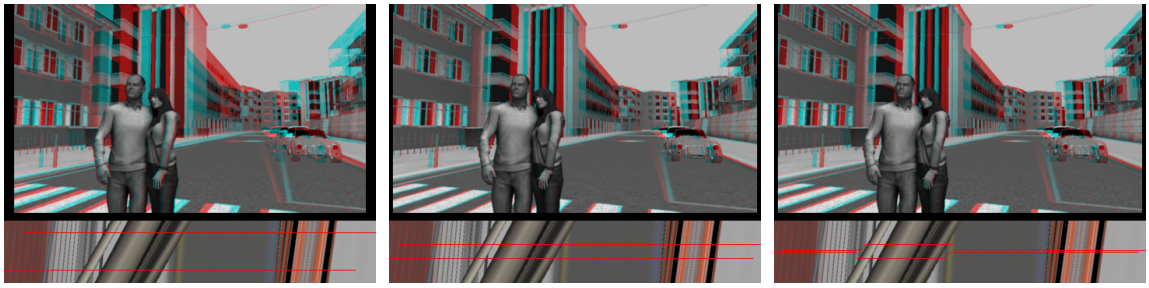


Figure 6.9: Comparison to Peleg et al. [2001]. Since the method of Peleg et al. supports only column-wise disparity control (*top*), it is not possible to achieve truly localized effects as with our per-pixel control over the disparity space (*bottom*).

6.2.6 Results

In this section we present results that are generated using our algorithm, given a prescribed set of disparity constraints. All results are presented as gray-scale, red-cyan anaglyph images (, red left). This not only allows for seeing the images stereoscopically in 3D, but also to quickly assess the horizontal parallax between images without glasses. We show results for computer-generated light fields as well as for real-world images, some of which are taken from UCSD/MERL Light Field Repository[1]. As high frame rate light field cameras are not yet available, we captured stop motion videos to test our method on live-action footage. For our results on computer-generated light fields, the normalized disparity volume D has been constructed from the z-buffer of the input images. For the real-world examples we used our depth reconstruction method presented in Chapter 4. We first provide a set of examples demonstrating the application of different disparity mapping operators φ. In our experiments, φ is defined on the normalized disparity, which is converted to pixel disparities by taking the physical depth budget, screen size and viewer position into account. For all results, we computed both output views.

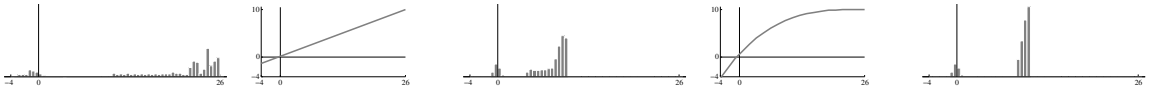
[1] http://graphics.ucsd.edu/datasets/lfarchive/



(a) Stereo pair with large baseline

(b) Stereo pair with small baseline

(c) Nonlinear depth enhancement of foreground



(d) Disparity histogram of (a)

(e) Mapping from (a) to (b)

(f) Disparity histogram of (b)

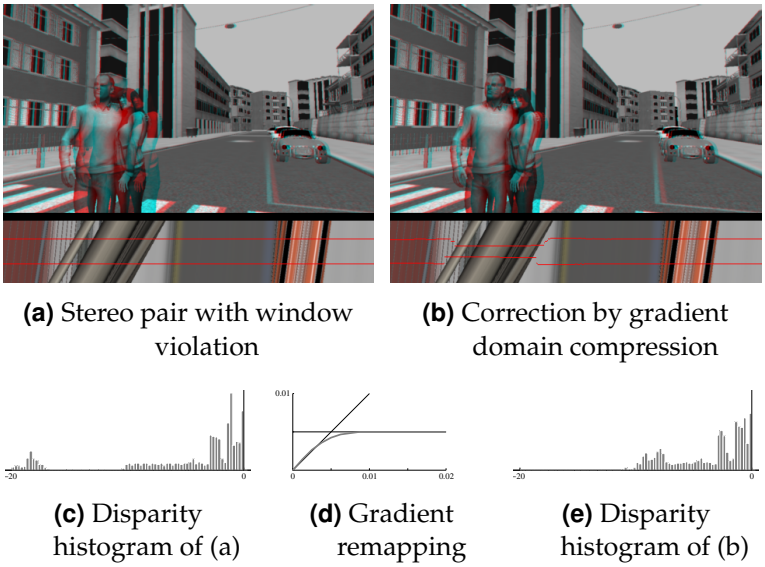
(g) Mapping from (a) to (c)

(h) Disparity histogram of (c)

Figure 6.10: Nonlinear disparity remapping. **(a)** shows a standard stereo pair with a large baseline where the foreground provides a good impression of depth. The background disparities, however, are quite large and can lead to ghosting artifacts or even the inability to fuse, when viewed on a larger screen. **(b)** Decreasing the baseline reduces the problems with respect to the background, but also considerably reduces the depth between the foreground and the background. **(c)** With a nonlinear disparity mapping function we can enhance the depth impression of the foreground, while keeping the maximum disparities in the background bounded as in **(b)**. Compare the disparity distribution **(h)** to that of the small baseline stereo **(f)**. **(d)**, **(f)** and **(h)** show the disparity distributions of respective stereo pairs, and **(e)** and **(g)** show the disparity remapping functions. Observe that the depth between the foreground and the background in **(d)** is preserved in **(h)**, while it is not in **(f)**. © 2011 Disney Enterprises, Inc.

Linear Remapping

The most straightforward example is a linear remapping of the disparity range, which corresponds to changing the camera baseline between two standard perspective views. In this case our method simply produces the expected planar cuts (e.g., Figure 6.10b). However, in this context a notable property of our method is that it eliminates the quite abstract and unintuitive concept of the camera baseline. Instead, one can directly specify the desired goal disparity range of the output images. This is the preferred method in actual production environments [Neuman, 2010].



(a) Stereo pair with window violation

(b) Correction by gradient domain compression

(c) Disparity histogram of (a)

(d) Gradient remapping

(e) Disparity histogram of (b)

Figure 6.11: Gradient domain compression. **(a)** A typical example for a stereo pair where the partially cropped couple features strong negative disparity, resulting in a so called window violation [Mendiburu, 2009]. Changing the convergence would increase the background disparities, potentially leading to the same problems as in Figure 6.10a. **(b)** With our method, we can resolve this stereoscopic issue by gradient domain compression of strong negative disparities. This effectively pushes the couple closer to the screen while keeping the background disparities unchanged. **(c)** and **(e)** show the disparity distribution of **(a)** and **(b)**, respectively. Note that in this example the empty space between the foreground and the background in **(c)** is squeezed in **(e)**. **(d)** shows the gradient remapping function, which effectively compresses strong disparity gradient. © 2011 Disney Enterprises, Inc.

Nonlinear and Gradient-Based Disparity Remapping

The strengths of our method are revealed for application scenarios where nonlinear changes of the disparity space are required. In principle, arbitrary remapping functions φ can be applied to construct the desired goal disparity volume, and even constant disparities are possible. For example, φ could be any of the nonlinear disparity mapping operators introduced by Lang et al. [2010] for display adaptation, stereoscopic error correction, or artistic effects. These functions can act globally on the complete domain as well as locally by remapping disparity gradients. For the gradient based remapping, we compute the gradient field in both u and v directions and process the gradients non-uniformly using the gradient magnitude, e.g., to supress big disparity jumps. We then reconstruct the height field by integration of the gradients using a Poisson solver [Agrawal and Raskar, 2007], and use this reconstructed height field to set a final goal disparity.

Figures 6.1, 6.10, 6.12, 6.11, and 6.14 show various examples for applications of nonlinear disparity remapping and gradient-based disparity processing. The



(a) Rhino, frame 43 **(b)** Rhino, frame 50 **(c)** Airplane, frame 100 **(d)** Airplane, frame 166

Figure 6.12: More examples for nonlinear disparity gradient remapping in order to reduce the overall disparity range, while preserving the perception of depth discontinuities and local depth variations. The first row shows the stereo pairs with two perspective images and a fixed baseline, while the second row shows the depth remapped versions. In particular, for the airplane scene the disparity gradient of the image's upper half was intensified, and the gradient of the lower half was attenuated. © 2011 Disney Enterprises, Inc.

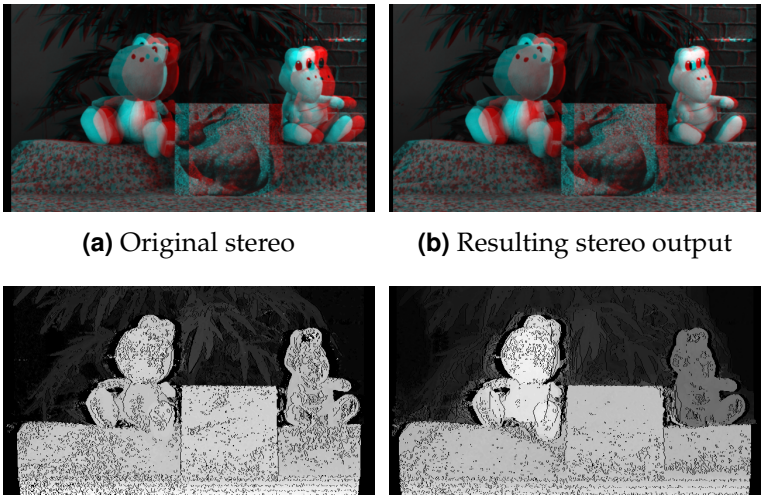
respectively used mapping function φ and the histograms of disparities before and after applying our method are shown as well. Figure 6.10 and 6.11 show typical issues arising in the production of stereoscopic content, and how they can be resolved using our method. In Figure 6.1 we use gradient domain remapping to increase the dramatic appearance of the scene by emphasizing depth gradients. Figure 6.12 shows a similar example where we reduce the overall disparity range while preserving the depth perception around depth discontinuities and local depth variations.

Artistic Control

In addition to the automatic mappings described above our method allows for concise manual control of disparities, which is an important requirement in any stereoscopic production environment. Users can directly modify the depth map $D(*,*,s)$ at the reference view s, e.g., using painting tools. The goal disparities are then set using the modified depth map. This allows for interesting artistic effects as well as fine-scale correction of the stereoscopic impression of a scene. Figure 6.9 and 6.13 show examples for manual control over disparity.

Multi-View Autostereoscopic Displays

A further interesting application domain is multi-view autostereoscopic displays. Similarly to stereoscopic displays, these displays have a limited depth budget.



(a) Original stereo **(b)** Resulting stereo output

(c) Disparity maps

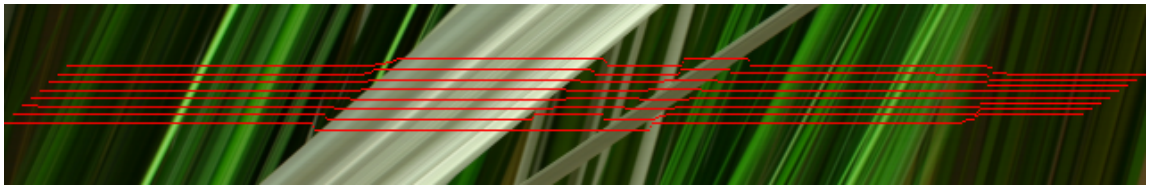
(d) Disparity histograms

Figure 6.13: Artistic control over depth. **(a)** and **(b)** We manually masked the two toys which are approximately at the same depth in the original stereo, and then defined different goal disparities for those regions. The resulting stereo output of our algorithm creates a different depth sensation for the two toys, even though they are placed at the same distance. **(c)** and **(d)** show the actual disparity maps and disparity distributions of the two stereoscopic image pairs. The two image pairs exhibit significantly different disparities in the area of two toys. Also note the new peak in the disparity histogram of the output stereo which corresponds to the toy on the right.

Thus, it is usually necessary to prefilter and remap an input light field to the available spatio-angular display bandwidth in order to avoid inter-perspective aliasing [Zwicker et al., 2006]. We can obtain properly remapped data to drive a particular automultiscopic display by computing multiple cuts through a light field. In Figure 6.14 we show an example for an 8-view autostereoscopic display from Alioscopy.

Performance

The computationally intensive steps of our method are the graph construction and the min-cut computation. The required time for the graph construction depends on the size of the underlying light field, while the optimization depends additionally on the complexity of the disparity constraints. The timings below have been



(a) Eight views optimized for a multi-view autostereoscopic display



(b) Photographs of these views shown on a 8-view autostereoscopic display

Figure 6.14: Multiple view generation. **(a)** Multi-perspective 8-view stereo, optimized with respect to the disparity range of an Alioscopy 8-view autostereoscopic display. **(b)** Unoptimized content easily leads to considerable ghosting artifacts (left column). Our method can automatically compute n-view stereo images that are designed to meet the disparity requirements of the output device and at the same time enhance perceived depth (right column).

measured for our MATLAB prototype on a machine with an Intel Core i7 2.8 Ghz CPU and 6 GB memory. For example, the complete graph construction for a single video frame consisting of 50 images with 640×480 resolution used for Figure 6.1 takes about 2.5 seconds. The min-cut computation with goal disparities computed by the gradient compression takes about 30 seconds. Overall per-frame computation times for all presented results ranged from 10 seconds to about 1.5 minutes for the more complex Airplane and Elephant data sets. In general, the more the goal disparities differ from the standard perspective of the input images the more processing time is required. The memory requirements of the employed graph cut algorithm for a light field of size $s \times u \times v$ are $115 \cdot s \cdot u \cdot v$ bytes. We reduce the memory footprint of our method via a variational formulation in Section 6.3. We also expect significant speed and memory improvements for more efficient graph-cut implementations.

Discussion

Our framework is very reliable when using dense light field sampling and accurate depth estimates. With a lower number of input images and less accurate depth map quality the cut may become less smooth and potentially cut through foreground regions. Fortunately, these effects can be compensated by setting λ and κ in Equation 6.4 higher to strengthen the depth and the radiance gradient based smoothness. By doing so, view transitions are less likely to happen inside well textured, foreground objects. With higher smoothness the output images are composed of standard perspective image segments which can also be interpreted as "multi-perspective image stitching."

Even when the scene is so complicated that it is very challenging to compute high quality depth, the proposed algorithm is quite robust and generally produces high quality output stereoscopic images. The rationale behind this is that for those regions where accurate cut computation is required to deal with the high frequency texture or the depth discontinuity, depth computation also becomes reliable for the very same reason. On the other hand, for those regions where the depth computation often becomes less reliable, such as texture-less regions with uniform color, the caused inaccuracy of the cut and thereby undesirable ray selection are not very noticeable as these regions are not visually salient.

6.3 Variational Formulation for View Synthesis

While the solution presented in the previous section provides a highly flexible and powerful control over disparity, one downside is that the employed graph cut optimization is memory intensive as it requires a dense regular graph structure to be built and maintained in memory. Additionally, the depth information has to be calculated and stored separately. These render it difficult to scale the method to higher resolution light fields. In this section we develop an alternative formulation based on variational optimization that avoids these problems. In the following, we treat a light field as a continuous function and derive an energy functional accordingly. Compared to the discrete formulation, the variational formulation runs with much less memory within comparable time.

6.3.1 Variational Formulation

As with the discrete formulation, the new formulation takes as input a light field and user-defined goal disparities. For a given reference view within the light field, it computes a new view such that the disparities between the two views best match

the prescribed goal disparities. As before, 3D light fields are assumed as input, and the goal disparities are also a 2D map, but defined at the *reference* view.

Let $\Omega \subset \mathbb{R}^2$ be the spatial domain of the (continuous) light field, and $\Gamma = [s_{\min}, s_{\max}] \subset \mathbb{R}$ be its bounded 1D angular domain. We then define a light field $L\colon [\Omega \times \Gamma] \to \mathbb{R}^3$, which maps a ray defined by a spatio-angular coordinate $(\mathbf{u}, s)$, where $\mathbf{u} = (u, v) \in \Omega$ and $s \in \Gamma$, to a sampled radiance represented in RGB color space. Further let $\hat{s} \in \Gamma$ denote the position of the reference image $I_{\hat{s}}(\mathbf{u}) = L(\mathbf{u}, \hat{s})$ for which the goal disparity map $G\colon \Omega \to \mathbb{R}$ is specified.

In the first step we shift the reference image $I_{\hat{s}}$ by the goal disparity G to obtain the *goal image*

$$I^*_{\hat{s}}(u + G(u,v), v) = I_{\hat{s}}(u,v). \tag{6.9}$$

The goal image $I^*_{\hat{s}}$ represents what the sought second view should look like. However, as the shifting is not injective nor surjective, there are ambiguities. We deal with the non-injectiveness that would map two pixels to the same location by selecting the pixel with the larger disparity, i.e., the one closer to the camera. To deal with the non-surjectiveness that leaves certain pixels without a disparity value, we mark these undefined regions in a binary mask $M\colon \Omega \to \{0, 1\}$ that is 0 in the undefined regions and 1 elsewhere. The undefined region is the disoccluded, monocular region which, in principle, should not be crucial to the depth perception, but may cause discomfort when conveying conflicting depth cues [Lang et al., 2010]. Many techniques fill this region by stretching neighboring image regions. However, this often introduces unwanted visible distortions of the image content.

Energy Functional

In our approach, we use pixels from the input light field to fill in information in these disoccluded regions. The unknown second view will hence be defined by a labeling function $l\colon \Omega \to \Gamma$ that determines for each pixel position in the second view, which input view a ray, i.e., a pixel, should be taken from. To find a smooth solution with least noticeable transitions (seams) we formulate the problem of finding l as a continuous optimization problem consisting of a data matching and a smoothness term

$$E(l) = \int_\Omega E_\mathrm{d}(l) \,+\, k\, E_\mathrm{s}(l)\; \mathrm{d}\mathbf{u}, \tag{6.10}$$

where $k > 0$ balances the two terms.

The data term E_d enforces the resulting second image to be as close as possible to the goal image in the subset of Ω where the goal image is defined, i.e., where

$M(\mathbf{u}) = 1$. Thus the data term is defined as

$$E_{\mathrm{d}}(l) = M(\mathbf{u}) \left\| L(\mathbf{u}, l(\mathbf{u})) - I^*_{\hat{s}}(\mathbf{u}) \right\|_1 . \tag{6.11}$$

The smoothness term E_{s} penalizes the amount of view transitions in the labeling. Importantly, it also guides the transitions to happen in less noticeable regions to allow for a seamless stitching of contributions from different images. For the disoccluded regions where the data term is disabled, the smoothness term allows for filling in information in a smooth manner, resulting in a least distorted completion of these missing regions. To achieve these goals, we define the smoothness term as the anisotropic total variation regularizer [Olsson et al., 2009; Grasmair and Lenzen, 2010]

$$E_{\mathrm{s}}(l) = \sqrt{\nabla l(\mathbf{u})^{\mathsf{T}} \, S(\mathbf{u}, l(\mathbf{u})) \, \nabla l(\mathbf{u})} \, . \tag{6.12}$$

The anisotropy is driven by the local variation in the light field, and measured using the structure tensor [Förstner and Gülch, 1987]

$$S(\mathbf{u}, s) = K_\sigma * (\nabla_{\mathbf{u}} L(\mathbf{u}, s) \, \nabla_{\mathbf{u}} L(\mathbf{u}, s)^{\mathsf{T}}), \tag{6.13}$$

where K_σ denotes a Gaussian kernel of variance σ^2, $*$ is the convolution operator, and $\nabla_{\mathbf{u}} L = (\partial_u L, \partial_v L)^{\mathsf{T}}$ is the spatial gradient of the light field L. The two orthonormal eigenvectors of S point along and across dominant spatial edges in the light field, respectively. Hence our smoothness term aligns view transitions with discontinuities in the light field which minimizes visible seam artifacts due to view transitions.

Substituting the energy terms in Equation 6.10 with Equations 6.11 and 6.12 results in the following variational minimization problem of the sought labeling l:

$$\min_l \int_\Omega M(\mathbf{u}) \left\| L(\mathbf{u}, l(\mathbf{u})) - I^*_{\hat{s}}(\mathbf{u}) \right\|_1 + k \sqrt{\nabla l(\mathbf{u})^{\mathsf{T}} \, S(\mathbf{u}, l(\mathbf{u})) \, \nabla l(\mathbf{u})} \, \mathrm{d}\mathbf{u}. \tag{6.14}$$

Convex Formulation

While the regularizer of the functional Equation 6.14 is convex, the data term is not. We reformulate Equation 6.14 as a convex functional using function lifting. We only outline the fundamental steps of the procedure here and refer to Pock et al. [2008] for more details.

Let us define a binary function $\phi \colon [\Omega \times \Gamma] \to \{0, 1\}$ with

$$\phi(\mathbf{u}, s) = \begin{cases} 1 & \text{if } l(\mathbf{u}) > s \\ 0 & \text{otherwise} \end{cases} , \tag{6.15}$$

which is the indicator for the s-superlevel sets of l. The feasible set of functions ϕ is

$$\mathcal{A}' = \{\phi \colon [\Omega \times \Gamma] \to \{0,1\} \mid \phi(\mathbf{u}, s_{\min}) = 1,\ \phi(\mathbf{u}, s_{\max}) = 0\}. \tag{6.16}$$

Rewriting Equation 6.14 with ϕ will now yield a convex data term, yet the feasible set of ϕ is non-convex and hence the minimization over it. To cope with this, ϕ is further relaxed so that it may take continuous values in the interval $[0,1]$, leading to the convex feasible set

$$\mathcal{A} = \{\phi \colon [\Omega \times \Gamma] \to [0,1] \mid \phi(\mathbf{u}, s_{\min}) = 1,\ \phi(\mathbf{u}, s_{\max}) = 0\}. \tag{6.17}$$

When ϕ is projected back to its original domain after the optimization, it is thresholded by some value within the interval $[0,1]$. The optimality is still guaranteed regardless the selection of threshold [Pock et al., 2008]. The labeling function l is recovered from ϕ by integrating over Γ [Chan et al., 2006]:

$$l(\mathbf{u}) = s_{\min} + \int_{\Gamma} \phi(\mathbf{u}, s)\, \mathrm{d}s. \tag{6.18}$$

Rewriting Equation 6.14 using the partial derivative of the indicator function ϕ, we obtain the following convex problem:

$$\min_{\phi \in \mathcal{A}} \int_{\Omega} \int_{\Gamma} M(\mathbf{u})\, \|L(\mathbf{u}, l(\mathbf{u})) - I_s^*(\mathbf{u})\|_1\, |\partial_s \phi(\mathbf{u}, s)| + k \sqrt{\nabla_{\mathbf{u}} \phi(\mathbf{u}, s)^{\mathsf{T}} S(\mathbf{u}, s) \nabla_{\mathbf{u}} \phi(\mathbf{u}, s)}\, \mathrm{d}s\, \mathrm{d}\mathbf{u}. \tag{6.19}$$

6.3.2 Optimization via Primal-Dual Iterations

A straightforward way to minimize the convex energy functional of Equation 6.19 would be to solve its associated Euler-Lagrange differential equation [Pock et al., 2008]. This appraoch is, however, complicated by the singularity of the used norms at zero. As an alternative we rewrite the norms in terms of their Wulff shape as the combined two norms constitute a convex, positively 1-homogeneous function [Zach et al., 2009].

Saddle-Point Formulation

A Wulff shape is defined as

$$W_{\phi} = \{\mathbf{y} \in \mathbb{R}^n \mid \langle \mathbf{y}, \mathbf{z} \rangle \leq \phi(\mathbf{z})\ \forall \mathbf{z} \in \mathbb{R}^n\}, \tag{6.20}$$

for a convex function $\phi\colon \mathbb{R}^n \to \mathbb{R}$ that is positively 1-homogeneous, i.e., $\phi(\lambda \mathbf{z}) = \lambda\, \phi(\mathbf{z})$, $\forall \lambda > 0$. It is a closed and bounded convex set containing zero, and is used to rewrite ϕ as

$$\phi(\mathbf{z}) = \max_{\mathbf{y} \in W_\phi} \langle \mathbf{z}, \mathbf{y} \rangle, \tag{6.21}$$

where the norms can be represented in a differentiable form. The minimization problem of Equation 6.19 can then be rewritten as

$$\min_{\phi \in \mathcal{A}} \max_{\mathbf{p} \in \mathcal{B}} E(\phi, \mathbf{p}), \tag{6.22}$$

with the energy functional

$$E(\phi, \mathbf{p}) \;=\; \int_\Omega \int_\Gamma \langle \nabla_{\mathbf{u},s}\, \phi(\mathbf{u}, s),\, \mathbf{p}(\mathbf{u}, s) \rangle \, \mathrm{d}s \, \mathrm{d}\mathbf{u}, \tag{6.23}$$

where $\nabla_{\mathbf{u},s}$ is now the gradient over all three dimensions of ϕ, and $\mathbf{p} = (\mathbf{p}_\mathbf{u}^\mathsf{T}, p_s) = (p_u, p_v, p_s)$ is the dual variable. The feasible set of the dual $\mathbf{p}$ then becomes the following Wulff shape:

$$\mathcal{B} = \left\{ \mathbf{p}\colon [\Omega \times \Gamma] \to \mathbb{R}^3 \;\middle|\; \sqrt{\mathbf{p}_\mathbf{u}^\mathsf{T}\, S(\mathbf{u}, s)\, \mathbf{p}_\mathbf{u}} \leq k,\, |p_s| \leq \rho(\mathbf{u}, s) \right\}, \tag{6.24}$$

where $\rho(\mathbf{u}, s)$ is the data term value at $(\mathbf{u}, s)$ as defined in Equation 6.11. This can be seen as a partial dualization in convex analysis, where ϕ is referred to as the primal variable and $\mathbf{p}$ the dual. Because we will maximize in the dual $\mathbf{p}$ and minimize in our original primal ϕ, the problem Equation 6.22 is called the saddle-point formulation.

Primal-Dual Iterations

To solve Equation 6.22, we alternate between taking gradient steps in the primal and dual [Handa et al., 2011]. To minimize the primal, we define the gradient as

$$\frac{\phi^n - \phi^{n+1}}{\sigma_p} = \nabla_\phi E(\phi, \mathbf{p}), \tag{6.25}$$

and to maximize the dual, we define the gradient as

$$\frac{\mathbf{p}^{n+1} - \mathbf{p}^n}{\sigma_p} = \nabla_\mathbf{p} E(\phi, \mathbf{p}). \tag{6.26}$$

By calculating the derivative of Equation 6.22 with respect to the primal and the dual we derive the update steps

$$\textbf{Primal:} \qquad \phi^{n+1} = \mathcal{P}_\mathcal{A}(\phi^n + \sigma_p \operatorname{div} \mathbf{p}^n) \tag{6.27}$$

$$\textbf{Dual:} \qquad \mathbf{p}^{n+1} = \mathcal{P}_\mathcal{B}(\mathbf{p}^n + \sigma_p \nabla \phi^{n+1}) \tag{6.28}$$

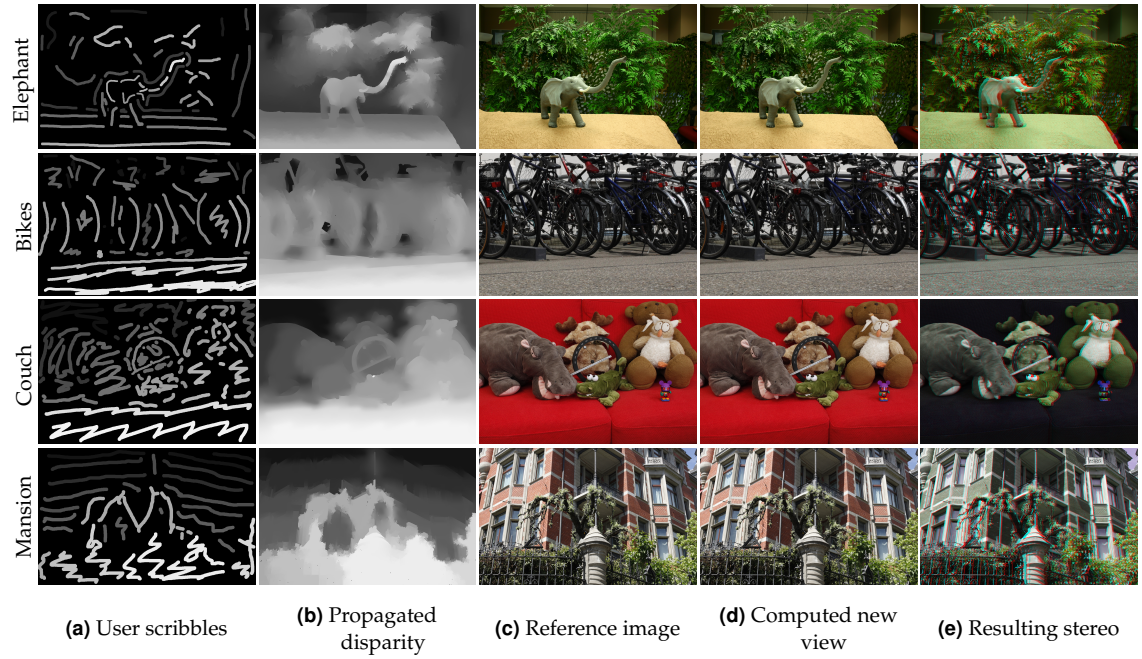


Figure 6.15: *Disparity modification using user scribbles.* This task demonstrates a possible use case, where sparse brush strokes are drawn by the user, **(a)**, and then propagated to form a dense goal disparity map **(b)**, from which the resulting stereo is generated. **(c)** and **(d)** show the reference view and the computed new view, respectively. **(e)** shows the resulting anaglyph stereo image. Note that the scribbles are not necessarily physically meaningful and are rather intended to test the flexibility and robustness of our method.

where $\mathcal{P}_\mathcal{A}$ projects ϕ back into its domain $\mathcal{A}$ by truncating it to $[0,1]$ and setting $\phi(\mathbf{u}, s_{\min}) = 1$ and $\phi(\mathbf{u}, s_{\max}) = 0$. $\mathcal{P}_\mathcal{B}$ is the Euclidean projector of the set $\mathcal{B}$:

$$\mathcal{P}_\mathcal{B}(\mathbf{p}^{n+1}) = \arg\min_{\mathbf{y}\in\mathcal{B}} \|\mathbf{p}^{n+1} - \mathbf{y}\|. \tag{6.29}$$

To compute the updates numerically, we discretize Ω and Γ so they represent pixel coordinates and the image index in the light field, respectively. The gradients are approximated using forward differences, but we use backward differences for the divergence to ensure convergence.

6.3.3 Experimental Results

In this section we evaluate the results from the variational formulation both qualitatively and quantitatively, also with the comparisons to the discrete formulation introduced in Section 6.2. We begin with the demonstration of two use cases: disparity modification using sparse user scribbles, and nonlinear disparity remapping. We then assess the results quantitatively, and finally analyze the performance. For

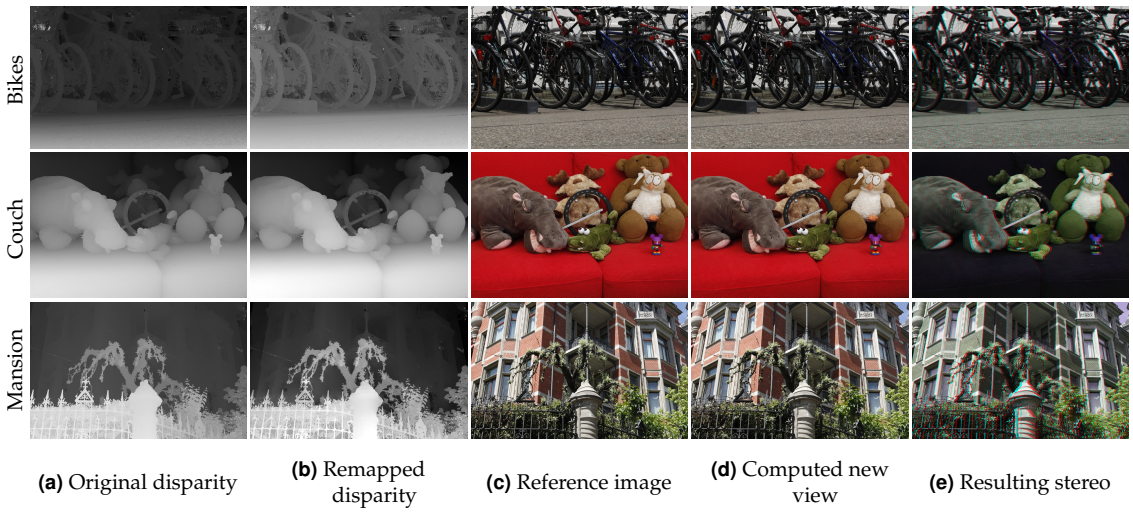


Figure 6.16: *Nonlinear disparity remapping.* The actual scene depth of the reference view **(a)** is nonlinearly remapped to create the goal disparity map **(b)**. For the Bikes dataset, the excessive disparity on the ground was compressed for a more comfortable stereoscopic viewing experience. For the Couch and Mansion datasets, the gradient of the disparity is modified such that large disparity gradients are removed, to better distribute the disparity budget and to obtain more local details. **(c)**–**(e)** show the reference image, the computed new view, and the resulting anaglyph stereo image, respectively.

all experiments, we used a fixed set of parameters, $\sigma_p = 1/\sqrt{3}$, $k = 10$, and $\sigma = 2$, and the primal-dual steps were iterated 10,000 times. All anaglyph images shown can be viewed in 3D using red-cyan anaglyph glasses as before.

Qualitative Evaluation

Our first use case based on *user scribbles* demonstrates a pipeline for the stereo editing and the 2D-to-3D conversion (see Figure 6.15). A sparse disparity annotation is provided by the user by drawing several brush strokes on top of the reference image, where the grayscale intensity of strokes encodes the amount of disparity. This sparse input is then propagated to form the dense goal disparity map using a method like StereoBrush [Wang et al., 2011]. Note that these scribbles need not necessarily be physically correct: our method finds the labeling that is closest to the specified disparity while producing the least noticeable seams, which leads to convincing stereo images. For instance, in the scribbles for the Couch dataset shown in Figure 6.15, the hippopotamus is pushed farther than all other stuffed animals, while it is the closest in reality.

Figure 6.16 shows the second use case, where the actual scene depth is *nonlinearly remapped* to convey modified depth perception. We used the scene depth computed using our reconstruction method in Chapter 4 for the Bikes, Couch, and Mansion



(a) Ordinary stereo **(b)** User scribbles **(c)** Nonlinear remapping

Figure 6.17: *Side-by-side comparisons between ordinary stereo and our results.* **(a)** shows an ordinary stereo consisting of two perspective images chosen from the input light field. **(b)** and **(c)** show our results using user scribbles and nonlinear disparity remapping, respectively.

datasets. For the Bikes dataset shown in the first row, the depth of the ground is compressed to give more disparity budget to the bikes at farther distance. For the Couch and Mansion datasets, the disparity gradient is obtained from the disparity map, and the high gradient is truncated to remove empty space and emphasize local details. The goal disparity map is then reconstructed from the modified gradient using a Poisson solver [Agrawal and Raskar, 2007].

Figure 6.17 presents the side-by-side comparisons of the "ordinary" stereo consisting of two perspective images, and our results using user scribbles and nonlinear remapping.

Quantitative Evaluation

To assess our results quantitatively we conducted two experiments where the expected results are known a priori. First, we use a single *constant disparity* value for all pixels as the goal disparity. Thus our formulation should result in the same image that is only translated by the amount of the disparity value, by best combining the pixels from the different views. Second, we use a *linearly scaled disparity map* of the reference view as the goal disparity. Since this linear scaling does not involve any local disparity modification, our method should choose the same, single input image entirely.

Figure 6.18 shows the results of constant disparity, where the goal disparity is set to

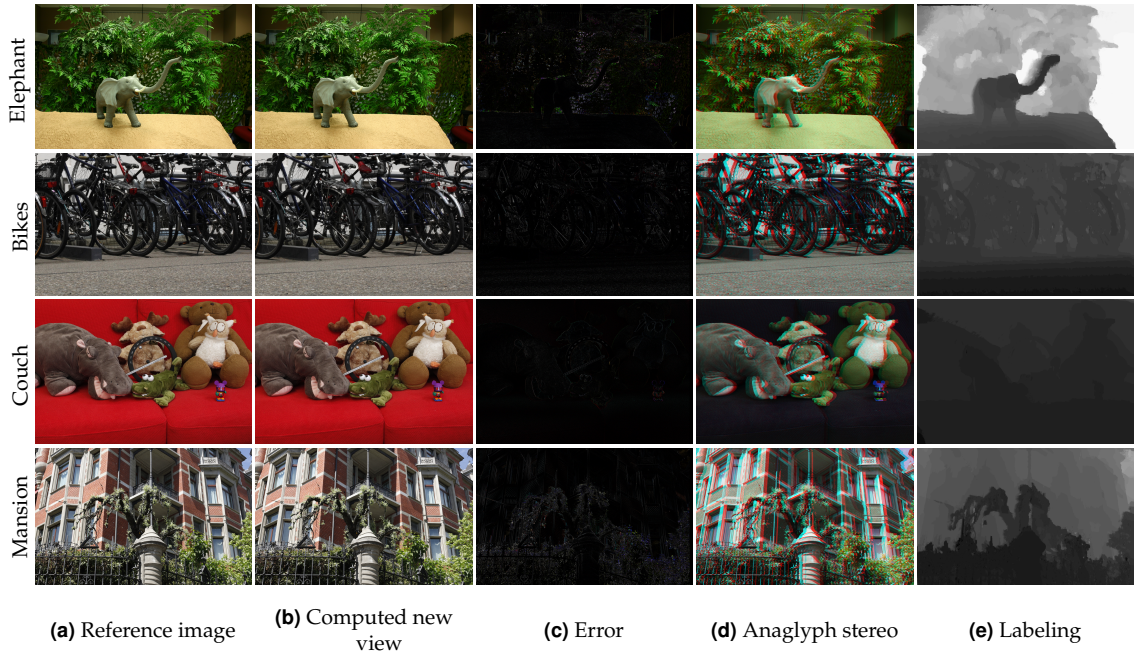


Figure 6.18: *Constant disparity.* **(a)** and **(b)** show the reference image and the computed new view given a fixed value of 20 pixels as the goal disparity. **(c)** shows the error of the computed image against the ground truth, for which we use the reference image translated by 20 pixels. The darker the pixel in the error image, the smaller the error. See Table 6.1 for corresponding RMSE measures. **(d)** shows the anaglyph stereo image, while **(e)** shows the resulting labeling. The resulting stereo should ideally look flat, but floating on the screen. The labeling images look like depth maps of the scenes. In fact, the rendering and the depth reconstruction problems are closely related; see Section 6.3.4.

20 pixels for all datasets. The third column shows the absolute difference between the image computed by our method and the reference image translated by the amount of disparity. The resulting anaglyph stereo images which are shown in the next column should look flat, but floating on the screen. The last column shows the resulting labeling, where each step in the grayscale denotes an image index. The labeling resembles the scene depth, and in fact, the stereoscopic rendering problem we are addressing and the dense depth estimation problem are tightly related. We discuss this in Section 6.3.4.

The results of a linearly scaled disparity are shown in Figure 6.19. We scaled the given depth map at the reference view by a factor of 10, hence each resulting view should equal to its tenth next view. As for the constant case, we show the reference image, the computed new view, and the difference between the computed image and the corresponding input image. The labeling shown in the last column should look close to flat for this experiment.

Table 6.1 lists the root-mean-squared errors (RMSE) of the computed views for the

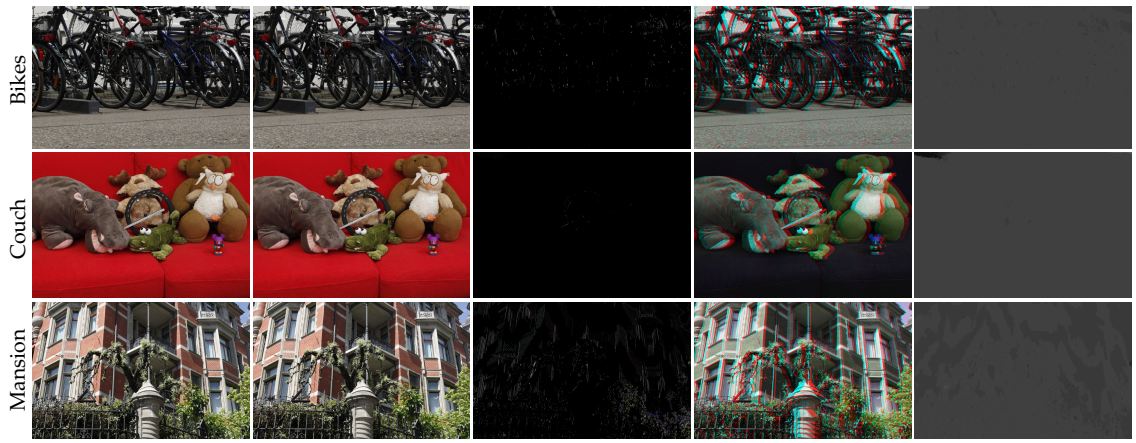
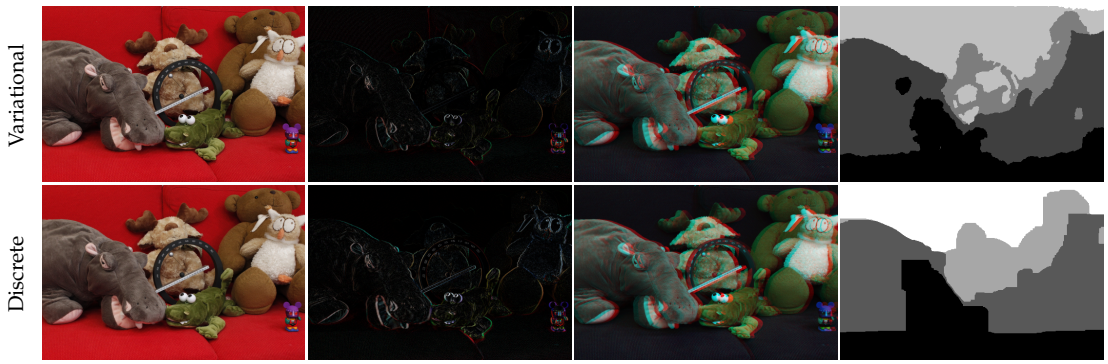


Figure 6.19: *Linear disparity scaling.* **(a)** and **(b)** show the reference image and the computed new view, for which the depth at the reference view was linearly scaled by a factor of 10 and used as the goal disparity map. **(c)** shows the error of the computed image against the ground truth, i.e., the 10th next image to the reference in the input light field. The darker the pixel in the error image, the smaller the error. See Table 6.1 for corresponding RMSE measures. **(d)** shows the anaglyph stereo image, and **(e)** shows the resulting labeling. The labeling should ideally look flat in this task.

two tasks against the ground truth, at two different resolutions: 1280×853 (1k) and 1920×1280 (2k). For both tasks of the Bikes and Couch datasets, the RMSE was all below 0.04. The error was higher for the Mansion dataset for both tasks, primarily due to the complex and thin structure of the tree and fence, which was about 0.07. Since the Elephant dataset is only available at 1k resolution we performed the constant disparity task at 1k resolution, which showed the RMSE of 0.05.

Goal disparity	RMSE from the ground truth			
	Elephant	Bikes	Couch	Mansion
1k resolution				
Constant	0.0499	0.0396	0.0235	0.0704
Linear	–*	0.0315	0.0072	0.0774
2k resolution				
Constant	–*	0.0392	0.0252	0.0735
Linear	–*	0.0288	0.0065	0.0693

Table 6.1: Errors of the computed views against the ground truth. (*Dataset not available)



(a) Computed view **(b)** Error **(c)** Stereo **(d)** Labeling

Figure 6.20: *Comparison to the discrete formulation.* We compare the constant disparity task (see Figure 6.18) against our discrete graph-cut formulation in Section 6.2. The labeling of the discrete formulation clearly shows grid bias, i.e., the transitions are mostly axis-aligned or diagonal (bottom **(d)**). This results in higher errors (bottom **(b)**).

Comparisons and Performance

A characteristic problem of the discrete formulations is that the optimization depends on the discretization, which is known as the grid bias. Figure 6.20 shows a side-by-side comparison of the constant disparity task between the results from the discrete and variational formulations. As seen in the labeling image, the discrete solver yields the labeling that is mostly aligned along the two image axes, and also exhibits a higher error in the final rendering.

We implemented the primal-dual iterations on a GPU using NVidia CUDA. The maximal GPU memory that the implementation requires at a time was measured for several different resolutions, both spatially and angularly. We show the memory footprint in Figure 6.21, also with the comparison to the discrete formulation of Section 6.2. The variational formulation uses less than 10 % of the memory compared to the discrete formulation. Table 6.2 summarizes the memory consumption of two formulations. We were not able to measure the memory footprint of the discrete formulation on 2k resolution datasets because the test system became unresponsive due to the excessive page swapping.

The running time varies depending on both the type of tasks and the light field resolution. Measured on an Intel Core i7 processor with 16 GB of RAM and an NVidia GTX 560 graphics card, the running time of the tasks for fifty 1k images varied between 10 and 12 minutes and for thirty 2k images between 13 and 15 minutes. Table 6.3 shows the complete running time of the variational formulation for all four tasks at two different resolutions.

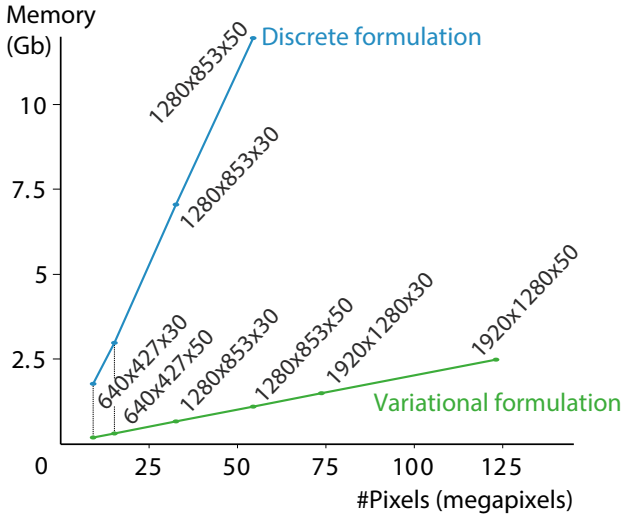


Figure 6.21: *Memory usage.* This graph shows the amount of memory that the two formulations of our method require for the input with different resolutions. Compared to the discrete formulation, the variational formulation is more memory efficient.

6.3.4 Relation to Depth Computation

The labeling results of the *constant disparity* task in Figure 6.18 resemble the actual scene depth. When a constant disparity value g is used as the goal disparity, i.e., $G(\mathbf{u}) = g, \forall \mathbf{u} \in \Omega$, the mapping between the reference image $I_{\hat{s}}$ and the goal image $I^*_{\hat{s}}$ in Equation 6.9 becomes bijective, with $M(\mathbf{u}) = 1$ everywhere except for the g-pixel-wide vertical strip at the left image border. The data term (Equation 6.11) can thus be rewritten as

$$E_\mathrm{d}(l) = \|L(u,v,l(u,v)) - L(u-g,v,\hat{s})\|_1\,, \tag{6.30}$$

for a pixel $\mathbf{u} = (u,v) \in \Omega : u > g$.

Minimizing this energy for l together with the smoothness term amounts to *correspondence matching*, i.e., finding the ray $(u,v,l(u,v))$ whose radiance best matches to that of $(u-g,v,\hat{s})$ for each pixel (u,v). The disparity is computed as

$$d = \frac{g}{l(u,v) - \hat{s}} \tag{6.31}$$

using simple triangulation. Since both g and $\hat{s}$ are constant, the labeling l encodes the disparity map.

In fact, the data term (Equation 6.11) implicitly implements dense disparity estimation, which can be shown more clearly using the *linear scaling* task. With the (scaled) actual disparity as the goal disparity one obtains a flat labeling as shown

Resolution (w×h×#images)	#pixels (Mpix)	Memory use in Mb		Ratio
		Variational	Discrete	
640×427×30	8.2	186.9	1,761.0	9.4%
640×427×50	13.7	310.0	2,983.4	9.2%
1k resolution				
1280×853×30	32.8	664.3	7,047.3	9.4%
1280×853×50	54.6	1,101.7	11,941.3	9.2%
2k resolution				
1920×1280×30	73.7	1,495.3	–*	–
1920×1280×50	122.9	2,479.7	–*	–

Table 6.2: Memory footprint of the variational formulation in comparison to the discrete formulation of Section 6.2. (*Test failed)

in Figure 6.19. Let us assume that the goal disparity G gives us an injective mapping from the reference image $I_{\hat{s}}$ to the goal image $I^*_{\hat{s}}$ in Equation 6.9. Substituting this mapping into the data term in Equation 6.11 yields

$$E_{\mathrm{d}}(l) = M(u,v) \parallel L(u,v,l(u,v)) - L(u - G(u,v), v, \hat{s}) \parallel_1 . \tag{6.32}$$

In our original problem, we fix the goal disparity G and seek the image index l for each pixel $(u,v) \in \Omega$. If, instead, we fix the labeling l to be a constant s' over Ω (i.e., flat labeling), and optimize the functional for G over all pixels $(u,v) \in \Omega$, the result will be the disparity map defined between the two images at the reference image $\hat{s}$ and the fixed other view s'. In this case the smoothness should accordingly be redefined in terms of G, instead of l.

6.4 Discussion

We have presented a general, multi-perspective framework for computing stereoscopic images from a light field, which satisfy a prescribed set of per-pixel goal disparity constraints. The core idea is to compute piecewise continuous cuts through the light field that minimize an energy derived from the goal disparities. We have demonstrated that our method is an effective and practical solution to key issues arising in today's stereoscopic content generation and post-production, and we believe that it will be an even more important tool for upcoming plenoptic cameras.

The presented framework provides a multitude of future research opportunities. For example, the current energy formulation strives at finding a cut that follows

Goal disparity	Computation time in seconds			
	Elephant	Bikes	Couch	Mansion
1k resolution				
Constant	§946	†513	660	614
Linear	–*	†518	667	623
Scribbles	636	†514	661	616
Remapping	–*	†516	669	628
2k resolution				
Constant	–*	838	863	803
Linear	–*	842	860	814
Scribbles	–*	840	858	799
Remapping	–*	837	860	805

Table 6.3: Running time of the variational formulation. We used 50 1k images or 30 2k images for the measurements. (*Dataset not available; §70 images used; †40 images used)

the goal disparity constraints as closely as possible without introducing visual artifacts. However, it could be valuable to extend this formulation with more sophisticated insights about stereoscopic perception, visual saliency, or temporal coherence. Moreover, our image generation selects pixels from the original input views and does not explicitly handle potential resampling issues. In this context, gradient-based image reconstruction, gradient-domain cuts, sub-pixel resolution techniques, and more sophisticated methods for up-sampling of light fields would be interesting to investigate. Finally, our solution to multiple cut generation defines goal disparities with respect to the reference view. To define disparity constraints for regions occluded in the reference view, this formulation could be extended to pairwise constraints between neighboring views.

On a more general level we would like to further investigate how our method relates to previous works such as Peleg et al. [2001] or Lang et al. [2010]. For instance, the stereoscopic image warping by Lang et al. [2010] could in principle be explained as planar cuts with a deformed light field or an adaptive *uv*-parameterization. We believe that a formulation of these techniques within our framework would lead to further interesting insights on stereoscopic imaging.

7

Conclusion

We presented a complete light field processing pipeline, from acquisition and geometry extraction to multiscopic rendering. Our algorithms are deliberately designed with the current advances of imaging hardware in mind, and work particularly well for light fields of high spatio-angular resolution.

7.1 Recapitulation

We began by presenting our acquisition setup that used a motorized linear stage driving the camera at sampling locations with accurate spacing. The sequence of images captured along the linear camera path forms a 3D light field, which is easy to capture and allows for efficient processing, while providing enough information for our key applications, namely, 3D geometry reconstruction and rendering. We also used unstructured light fields captured by a hand-held camera, for which required preprocessing steps were explained.

We pointed out the advantage of high angular resolution, which makes correspondence matching significantly more robust and precise, while matching over larger patches is avoided. Based on this, we described a depth computation algorithm which is tailored towards light fields of high resolution both spatially and angularly. Due to its small memory footprint and highly localized computation, our algorithm can be efficiently implemented on the GPU. We avoid costly global optimizations by resorting to the novel hierarchical scheme that we call fine-to-coarse refinement. The quality of our reconstructions is supported by extensive evaluations. We

presented a few applications including segmentation and image-based rendering as well as 3D reconstruction.

We then focused on the analysis of the density of a light field required for high quality 3D reconstruction. We developed a sampling analysis model targeted at geometry reconstruction from light fields, which provided an answer to the question of finding best sampling locations. This is achieved by analyzing the visibility of scene points and the quality of correspondence matching, and estimating the distribution of reliable depth estimates over sampling locations.

Having a dense light field and the accurate depth for each ray allows for a new paradigm for 3D rendering with great controllability over perceived depth, which is well suited for stereoscopic post-processing. We developed a method which directly renders a pair of stereoscopic images from a light field and allows us to specify the desired disparity on a per-pixel basis. Given this desired disparity map, it calculates non-planar 2D cuts within the volume of the light field, which balance between the fidelity and smoothness. The images are sampled from the cut surfaces and essentially possess multiple perspectives. We presented two formulations based on energy minimization to solve for these cuts. By computing more than two cuts, the framework is easily extended to multi-view displays.

7.2 Limitations and Future Work

Our algorithms open up several opportunities for improvements and future research.

7.2.1 Geometry Reconstruction

One of the limitations of our reconstruction method is that currently it does not provide an explicit treatment for dynamic scenes. Simply reconstructing the scene frame by frame may lead to an unsatisfactory result perhaps due to temporal flickering, and also means losing chances to use additional correspondence information over time. A significant advantage would be gained if the current reconstruction method could be extended to the temporal domain, e.g., by implementing correspondence matching and regularization over time, while remaining still efficient.

As shown in Figure 4.14, our depth reconstruction algorithm has difficulties when dealing with noisy, low (especially, angular) resolution input. Although not targeted at such input, it would be more desirable that the algorithm fails more gracefully. While the noisy input is not as substantial a problem as before thanks to the progress of sensor technologies, it is still expensive to build an array including

dozens of cameras. Thus, a higher priority could be put on the adaptability to lower angular resolution input.

Since our depth values are discrete, reconstructed depth maps are essentially composed of many fronto-parallel surfaces, which could be noticeable when viewed in greater details. Either a filter to reduce the discreteness or a scheme to get continuous depth values would be promising. In general, a filtering mechanism with a more principled approach than our current bilateral median filter would also be a plus.

Lastly, our method does not handle the surfaces with varying directional reflectance, such as specular objects. A number of methods have been proposed to remove specular components before estimating the depth using only the albedo, such as Criminisi et al. [2005], while others attempt to jointly estimate both the geometry and the reflectance, such as Goldman et al. [2010]. We believe the high angular resolution will also support greatly the identification of specularity or the joint estimation of the reflectance function (BRDF) and the geometry of the surface. Another solution could be to avoid highly specular regions in the angular domain for depth estimation, which could be implemented as a component of the preference function used in our analysis model.

7.2.2 Sampling Analysis

While our reconstruction method can be extended for 4D and unstructured light fields, our analysis model cannot yet. Having our analysis applicable to other types of light field could be among the most fruitful extensions to our sampling analysis model.

We used the same, empirically chosen fixed value for the accuracy function $\mathrm{Acc}(\mathcal{C})$ in the preference function for depth resolvability (Equation 5.4). In principle, this should be estimated for each correspondence matching algorithm. It would also be valuable to investigate various sources of information that can be used to augment and estimate the preference function $\gamma(s)$ in Equation 5.5, such as view-dependent effects, illumination changes, monocular depth cues, or semantic information like annotations.

Ultimately, an interactive system with live feedback to scan a scene would be of great use. The system monitors the sampling rate required for the 3D reconstruction and guides the user to scan under-sampled regions of the light field, similar to what Davis et al. [2012] did for rendering.

7.2.3 Rendering

Although 1D angular variation of a light field is enough to create horizontal parallax required for most 3D displays, 2D variation (as in 4D light fields) could provide more flexibility and expressiveness. An example is finite depth of field. Currently, our method generates pinhole images without any defocus blur (unless the input already had it). Finite depth of field is an important photographic technique for artistic expressions, and natural and well-shaped defocus blur (often called *Bokeh*) would require an integration of rays over a 2D region of a light field.

Another possible improvement would be to handle sparse light fields. The resolution of the modifiable disparity of a pixel depends on the minimum disparity that is captured in the light field, i.e., the disparity of the pixel between two adjacent views in the light field. A disparity finer than this may not be achieved, simply because the ray coming from the scene point with the wanted disparity does not exist in the light field that we take rays from. Thus, the adjustment is possible only up to an integer multiple of this minimum disparity, and cannot be done for a fraction of it. A solution could be implemented via a type of interpolation over the angular dimension, relaxing the range of the labeling function $l(\mathbf{u})$ from an integer to a real number.

While our variational formulation remedies extensive memory requirements of the graph cut optimization for high resolution input, it is still time expensive. A truly useful tool would provide an interactive feedback to changing goals, parameters, etc. Although the variational formulation is already one step closer to such favorable properties since it works iteratively and intermediate steps can be displayed for visual feedback, it still is far from an interactive rate. An approximate formulation or faster solver would be much fruitful in this regard.

References

[Adelson and Bergen, 1991] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, pages 3–20, 1991.

[Adelson and Wang, 1992] Edward H. Adelson and John Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106, 1992.

[Agrawal and Raskar, 2007] Amit Agrawal and Ramesh Raskar. Gradient domain manipulation techniques in vision and graphics. In *Courses at IEEE International Conference on Computer Vision*, 2007.

[Ayvaci et al., 2012] Alper Ayvaci, Michalis Raptis, and Stefano Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision*, 97(3):322–338, 2012.

[Basha et al., 2012] Tali Basha, Shai Avidan, Alexander Hornung, and Wojciech Matusik. Structure and motion from scene registration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433, 2012.

[Beeler et al., 2010] Thabo Beeler, Bernd Bickel, Paul A. Beardsley, Robert W. Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics*, 29(4):40:1–40:9, 2010.

[Beeler et al., 2011] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul A. Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. High-

quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics*, 30(4):75:1–75:10, 2011.

[Bishop and Favaro, 2010] Tom E. Bishop and Paolo Favaro. Full-resolution depth map estimation from an aliased plenoptic light field. In *Proceedings of Asian Conference on Computer Vision*, pages 186–200, 2010.

[Bishop et al., 2009] Tom E. Bishop, Sara Zanetti, and Paolo Favaro. Light field superresolution. In *Proceedings of IEEE International Conference on Computational Photography*, pages 1–9, 2009.

[Bleyer et al., 2011] Michael Bleyer, Carsten Rother, Pushmeet Kohli, Daniel Scharstein, and Sudipta Sinha. Object stereo — Joint stereo matching and object segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3081–3088, 2011.

[Bolles and Baker, 1987] Robert C. Bolles and H. Harlyn Baker. Epipolar-plane image analysis: A technique for analyzing motion sequences. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 26–36, 1987.

[Bolles et al., 1987] Robert C. Bolles, H. Harlyn Baker, and David H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[Bowles et al., 2012] Huw Bowles, Kenny Mitchell, Robert W. Sumner, Jeremy Moore, and Markus Gross. Iterative image warping. *Computer Graphics Forum*, 31(2):237–246, 2012.

[Boykov and Kolmogorov, 2004] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[Boykov et al., 2001] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[Buehler et al., 2001] Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler, and Michael F. Cohen. Unstructured Lumigraph rendering. In *Proceedings of ACM SIGGRAPH*, pages 425–432, 2001.

[Čech and Šára, 2007] Jan Čech and Radim Šára. Efficient sampling of disparity space for fast and accurate matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[Chai et al., 2000] Jinxiang Chai, Shing-Chow Chan, Heung-Yeung Shum, and Xin

Tong. Plenoptic sampling. In *Proceedings of ACM SIGGRAPH*, pages 307–318, 2000.

[Chan et al., 2006] Tony F. Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics*, 66(5):1632–1648, 2006.

[Chang et al., 2011] Che-Han Chang, Chia-Kai Liang, and Yung-Yu Chuang. Content-aware display adaptation and interactive editing for stereoscopic images. *IEEE Transactions on Multimedia*, 13(4):589–601, 2011.

[Chaurasia et al., 2013] Gaurav Chaurasia, Sylvain Duchêne, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics*, 32(3):30:1–30:12, 2013.

[Chen et al., 2002] Wei-Chao Chen, Jean-Yves Bouguet, Michael H. Chu, and Radek Grzeszczuk. Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Transactions on Graphics*, 21(3):447–456, 2002.

[Comaniciu and Meer, 2002] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[Criminisi et al., 2005] Antonio Criminisi, Sing Bing Kang, Rahul Swaminathan, Richard Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer Vision and Image Understanding*, 97(1):51–85, 2005.

[Davis et al., 2012] Abe Davis, Marc Levoy, and Frédo Durand. Unstructured light fields. *Computer Graphics Forum*, 31(2):305–314, 2012.

[Debevec et al., 1996] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of ACM SIGGRAPH*, pages 11–20, 1996.

[Didyk et al., 2010] Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. Adaptive image-space stereo view synthesis. In *Proceedings of International Symposium on Vision, Modeling and Visualization*, pages 299–306, 2010.

[Didyk et al., 2011] Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. A perceptual model for disparity. *ACM Transactions on Graphics*, 30(4):96:1–96:10, 2011.

[Didyk et al., 2012a] Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. Apparent stereo: The Cornsweet illu-

sion can enhance perceived depth. In *Proceedings of SPIE, Volume 8291, Human Vision and Electronic Imaging XVII*, pages 0N:1–12, 2012.

[Didyk et al., 2012b] Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, Hans-Peter Seidel, and Wojciech Matusik. A luminance-contrast-aware disparity model and applications. *ACM Transactions on Graphics*, 31(6):184:1–184:10, 2012.

[Didyk et al., 2013] Piotr Didyk, Pitchaya Sitthi-amorn, William T. Freeman, Frédo Durand, and Wojciech Matusik. Joint view expansion and filtering for automultiscopic 3d displays. *ACM Transactions on Graphics*, 32(6):221:1–221:8, 2013.

[Duda et al., 1995] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification and Scene Analysis, 2nd Edition*. Wiley Interscience, 1995.

[Durand et al., 2005] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. *ACM Transactions on Graphics*, 24(3):1115–1126, 2005.

[Egan et al., 2011] Kevin Egan, Florian Hecht, Frédo Durand, and Ravi Ramamoorthi. Frequency analysis and sheared filtering for shadow light fields of complex occluders. *ACM Transactions on Graphics*, 30(2):9:1–9:13, 2011.

[Fitzgibbon et al., 2005] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005.

[Förstner and Gülch, 1987] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proceedings of ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, 1987.

[Furukawa and Ponce, 2010] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[Furukawa et al., 2010] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Towards Internet-scale multi-view stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1434–1441, 2010.

[Fusiello et al., 2000] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[Gallup et al., 2008] David Gallup, Jan-Michael Frahm, Philippos Mordohai, and

Marc Pollefeys. Variable baseline/resolution stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[Geiger et al., 2010] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Proceedings of Asian Conference on Computer Vision*, pages 25–38, 2010.

[Georgiev and Lumsdaine, 2010] Todor Georgiev and Andrew Lumsdaine. Reducing plenoptic camera artifacts. *Computer Graphics Forum*, 29(6):1955–1968, 2010.

[Georgiev et al., 2006] Todor Georgiev, Ke Colin Zheng, Brian Curless, David Salesin, Shree Nayar, and Chintan Intwala. Spatio-angular resolution tradeoffs in integral photography. In *Proceedings of Eurographics Symposium on Rendering*, pages 263–272, 2006.

[Goesele et al., 2007] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[Goldlücke and Magnor, 2003] Bastian Goldlücke and Marcus Magnor. Joint 3D-reconstruction and background separation in multiple views using graph cuts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 683–688, 2003.

[Goldman et al., 2010] Dan B. Goldman, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2010.

[Gortler et al., 1996] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The Lumigraph. In *Proceedings of ACM SIGGRAPH*, pages 43–54, 1996.

[Grasmair and Lenzen, 2010] Markus Grasmair and Frank Lenzen. Anisotropic total variation filtering. *Applied Mathematics & Optimization*, 62(3):323–339, 2010.

[Grossmann, 1987] P. Grossmann. Depth from focus. *Pattern Recognition Letters*, 5(1):63–69, 1987.

[Gupta and Hartley, 1997] Rajiv Gupta and Richard I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, 1997.

[Handa et al., 2011] Ankur Handa, Richard A. Newcombe, Adrien Angeli, and Andrew J. Davison. Applications of Legendre-Fenchel transformation to com-

puter vision problems. Technical Report DTR11-7, Imperial College, Department of Computing, 2011.

[Heber and Pock, 2014] Stefan Heber and Thomas Pock. Shape from light field meets robust PCA. In *Proceedings of European Conference on Computer Vision*, pages 751–767, 2014.

[Heinzle et al., 2011] Simon Heinzle, Pierre Greisen, David Gallup, Christine Chen, Daniel Saner, Aljoscha Smolic, Andreas Burg, Wojciech Matusik, and Markus Gross. Computational stereo camera system with programmable control loop. *ACM Transactions on Graphics*, 30(4):94:1–94:10, 2011.

[Hirschmüller, 2005] Heiko Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 807–814, 2005.

[Holliman, 2004] Nicolas Holliman. Mapping perceived depth to regions of interest in stereoscopic images. In *Proceedings of SPIE, Volume 5291, Stereoscopic Displays and Applications XV*, pages 117–128, 2004.

[Hornung and Kobbelt, 2009] Alexander Hornung and Leif Kobbelt. Interactive pixel-accurate free viewpoint rendering from images with silhouette aware sampling. *Computer Graphics Forum*, 28(8):2090–2103, 2009.

[Hornung et al., 2008] Alexander Hornung, Boyi Zeng, and Leif Kobbelt. Image selection for improved multi-view stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[Humayun et al., 2011] Ahmad Humayun, Oisin Mac Aodha, and Gabriel J. Brostow. Learning to find occlusion regions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2011.

[Isaksen et al., 2000] Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of ACM SIGGRAPH*, pages 297–306, 2000.

[Jones et al., 2001] Graham Jones, Delman Lee, Nicolas Holliman, and David Ezra. Controlling perceived depth in stereoscopic images. In *Proceedings of SPIE, Volume 4297, Stereoscopic Displays and Virtual Systems VIII*, pages 42–53, 2001.

[Joo et al., 2014] Hanbyul Joo, Hyun Soo Park, and Yaser Sheikh. MAP visibility estimation for large-scale dynamic 3D reconstruction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1122–1129, 2014.

[Joshi et al., 2006] Neel Joshi, Wojciech Matusik, and Shai Avidan. Natural video matting using camera arrays. *ACM Transactions on Graphics*, 25(3):779–786, 2006.

[Kanade et al., 1997] Takeo Kanade, Peter Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):34–47, 1997.

[Kang and Szeliski, 2004] Sing Bing Kang and Richard Szeliski. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision*, 58(2):139–163, 2004.

[Kolmogorov and Zabih, 2001] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions via graph cuts. In *Proceedings of IEEE International Conference on Computer Vision*, pages 508–515, 2001.

[Koppal et al., 2011] Sanjeev J. Koppal, C. Lawrence Zitnick, Michael F. Cohen, Sing Bing Kang, Bryan Ressler, and Alex Colburn. A viewer-centric editor for 3D movies. *IEEE Computer Graphics and Applications*, 31(1):20–35, 2011.

[Krainin et al., 2011] Michael Krainin, Brian Curless, and Dieter Fox. Autonomous generation of complete 3D object models using next best view manipulation planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 5031–5037, 2011.

[Kutulakos and Dyer, 1994] Kiriakos N. Kutulakos and Charles R. Dyer. Recovering shape by purposive viewpoint adjustment. *International Journal of Computer Vision*, 12(2-3):113–136, 1994.

[Lang et al., 2010] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross. Nonlinear disparity mapping for stereoscopic 3D. *ACM Transactions on Graphics*, 29(4):75:1–75:10, 2010.

[Lang et al., 2012] Manuel Lang, Oliver Wang, Tunc Aydin, Aljoscha Smolic, and Markus Gross. Practical temporal consistency for image-based graphics applications. *ACM Transactions on Graphics*, 31(4):34:1–34:8, 2012.

[Lanman et al., 2011] Douglas Lanman, Gordon Wetzstein, Matthew Hirsch, Wolfgang Heidrich, and Ramesh Raskar. Polarization fields: Dynamic light field display using multi-layer LCDs. *ACM Transactions on Graphics*, 30(6):186:1–186:10, 2011.

[Levin and Durand, 2010] Anat Levin and Frédo Durand. Linear view synthesis using a dimensionality gap light field prior. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1838, 2010.

[Levin et al., 2008a] Anat Levin, William T. Freeman, and Frédo Durand. Understanding camera trade-offs through a Bayesian analysis of light field projections. In *Proceedings of European Conference on Computer Vision*, pages 88–101, 2008.

[Levin et al., 2008b] Anat Levin, Peter Sand, Taeg Sang Cho, Frédo Durand, and William T. Freeman. Motion-invariant photography. *ACM Transactions on Graphics*, 27(3):71:1–71:9, 2008.

[Levin et al., 2009] Anat Levin, Samuel W. Hasinoff, Paul Green, Frédo Durand, and William T. Freeman. 4D frequency analysis of computational cameras for depth of field extension. *ACM Transactions on Graphics*, 28(3):97:1–97:14, 2009.

[Levoy and Hanrahan, 1996] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of ACM SIGGRAPH*, pages 31–42, 1996.

[Liang et al., 2008] Chia-Kai Liang, Tai-Hsu Lin, Bing-Yi Wong, Chi Liu, and Homer H. Chen. Programmable aperture photography: Multiplexed light field acquisition. *ACM Transactions on Graphics*, 27(3):55:1–55:10, 2008.

[Lin and Shum, 2000] Zhouchen Lin and Heung-Yeung Shum. On the number of samples needed in light field rendering with constant-depth assumption. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2000.

[Lippmann, 1908a] Gabriel Lippmann. Épreuves réversibles donnant la sensation du relief. *Journal de Physique Théorique et Appliquée*, 7(1):821–825, 1908.

[Lippmann, 1908b] Gabriel Lippmann. Épreuves réversibles. photographies intégrales. *Comptes Rendus de l'Académie des Sciences*, 146(9):446–451, 1908.

[Masia et al., 2013a] Belen Masia, Gordon Wetzstein, Carlos Aliaga, Ramesh Raskar, and Diego Gutierrez. Display adaptive 3D content remapping. *Computers & Graphics*, 37(8):983–996, 2013.

[Masia et al., 2013b] Belen Masia, Gordon Wetzstein, Piotr Didyk, and Diego Gutierrez. A survey on computational displays: Pushing the boundaries of optics, computation, and perception. *Computers & Graphics*, 37(8):1012–1038, 2013.

[Matusik and Pfister, 2004] Wojciech Matusik and Hanspeter Pfister. 3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Transactions on Graphics*, 23(3):814–824, 2004.

[Maver and Bajcsy, 1993] Jasna Maver and Ruzena Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–433, 1993.

[Mendiburu, 2009] Bernard Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009.

[Neuman, 2010] Robert Neuman. Personal Communication with Robert Neuman, Chief Stereographer, Disney Animation Studios, 2010.

[Ng et al., 2005] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. Technical Report CSTR 2005-02, Stanford University, 2005.

[Nomura et al., 2007] Yoshikuni Nomura, Li Zhang, and Shree K. Nayar. Scene collages and flexible camera arrays. In *Proceedings of Eurographics Symposium on Rendering*, pages 127–138, 2007.

[Okutomi and Kanade, 1993] Masatoshi Okutomi and Takeo Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.

[Olague and Mohr, 2002] Gustavo Olague and Roger Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognition*, 35(4):927–944, 2002.

[Olsson et al., 2009] Carl Olsson, Martin Byröd, Niels Chr. Overgaard, and Fredrik Kahl. Extending continuous cuts: Anisotropic metrics and expansion moves. In *Proceedings of IEEE International Conference on Computer Vision*, pages 405–412, 2009.

[Oskam et al., 2011] Thomas Oskam, Alexander Hornung, Huw Bowles, Kenny Mitchell, and Markus Gross. OSCAM — Optimized stereoscopic camera control for interactive 3D. *ACM Transactions on Graphics*, 30(6):189:1–189:8, 2011.

[Pajdla, 2002] Tomas Pajdla. Geometry of two-slit camera. Research Report CTU-CMP-2002-02, Czech Technical University, 2002.

[Peleg et al., 2001] Shmuel Peleg, Moshe Ben-Ezra, and Yael Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):279–290, 2001.

[Pentland, 1987] Alex Paul Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):523–531, 1987.

[Pock et al., 2008] Thomas Pock, Thomas Schoenemann, Gottfried Graber, Horst Bischof, and Daniel Cremers. A convex formulation of continuous multi-label problems. In *Proceedings of European Conference on Computer Vision*, pages 792–805, 2008.

[Rademacher and Bishop, 1998] Paul Rademacher and Gary Bishop. Multiple-center-of-projection images. In *Proceedings of ACM SIGGRAPH*, pages 199–206, 1998.

[Rhemann et al., 2011] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3017–3024, 2011.

[Rubinstein et al., 2008] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Improved seam carving for video retargeting. *ACM Transactions on Graphics*, 27(3):16:1–16:9, 2008.

[Scharstein and Szeliski, 2002] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[Schechner and Kiryati, 2000] Yoav Y. Schechner and Nahum Kiryati. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision*, 39(2):141–162, 2000.

[Scott et al., 2003] William R. Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys*, 35(1):64–96, 2003.

[Seitz and Dyer, 1999] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[Seitz et al., 2006] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[Seitz, 2001] Steve Seitz. The space of all stereo images. In *Proceedings of IEEE International Conference on Computer Vision*, pages 26–33, 2001.

[Shibata et al., 2011] Takashi Shibata, Joohwan Kim, David M. Hoffman, and Martin S. Banks. The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of Vision*, 11(8):11, 1–29, 2011.

[Shum and Kang, 2000] Heung-Yeung Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Proceedings of SPIE, Volume 4067, Visual Communications and Image Processing 2000*, pages 2–13, 2000.

[Shum et al., 2007] Heung-Yeung Shum, Shing-Chow Chan, and Sing Bing Kang. *Image-Based Rendering*. Springer, 2007.

[Snavely et al., 2008] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from Internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[Soukup et al., 2014] Daniel Soukup, Reinhold Huber-Mörk, Svorad Štolc, and Branislav Holländer. Depth estimation within a multi-line-scan light-field framework. In *Proceedings of International Symposium on Advances in Visual Computing*, pages 471–481, 2014.

[Stich et al., 2006] Timo Stich, Art Tevs, and Marcus Magnor. Global depth from epipolar volumes — A general framework for reconstructing non-Lambertian surfaces. In *International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 916–923, 2006.

[Sun et al., 2011] Xun Sun, Xing Mei, Shaohui Jiao, Mingcai Zhou, and Haitao Wang. Stereo matching with reliable disparity propagation. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 132–139, 2011.

[Sylwan, 2010] Sebastian Sylwan. The application of vision algorithms to visual effects production. In *Proceedings of Asian Conference on Computer Vision*, pages 189–199, 2010.

[Szeliski and Scharstein, 2002] Richard Szeliski and Daniel Scharstein. Symmetric sub-pixel stereo matching. In *Proceedings of European Conference on Computer Vision*, pages 525–540, 2002.

[Szeliski and Scharstein, 2003] Richard Szeliski and Daniel Scharstein. Sampling the disparity space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):419–425, 2003.

[Tao et al., 2013] Michael W. Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of IEEE International Conference on Computer Vision*, pages 673–680, 2013.

[Thomas and Johnston, 1995] Frank Thomas and Ollie Johnston. *The Illusion of Life: Disney Animation*. Hyperion, Los Angeles, 1995.

[Vaish et al., 2005] Vaibhav Vaish, Gaurav Garg, Eino-Ville Talvala, Emilio Antunez, Bennett Wilburn, Mark Horowitz, and Marc Levoy. Synthetic aperture focusing using a shear-warp factorization of the viewing transform. In *Proceedings of Workshop on Advanced 3D Imaging for Safety and Security*, 2005.

[Vaish et al., 2006] Vaibhav Vaish, Marc Levoy, Richard Szeliski, C. Lawrence Zitnick, and Sing Bing Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2331–2338, 2006.

[Vázquez et al., 2003] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. Automatic view selection using viewpoint entropy and its applications to image-based modelling. *Computer Graphics Forum*, 22(4):689–700, 2003.

[Veeraraghavan et al., 2007] Ashok Veeraraghavan, Ramesh Raskar, Amit K. Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: Mask en-

hanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics*, 26(3):69:1–69:12, 2007.

[Venkataraman et al., 2013] Kartik Venkataraman, Dan Lelescu, Jacques Duparré, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis, and Shree Nayar. PiCam: An ultra-thin high performance monolithic camera array. *ACM Transactions on Graphics*, 32(6):166:1–166:13, 2013.

[Vu et al., 2009] Hoang-Hiep Vu, Renaud Keriven, Patrick Labatut, and Jean-Philippe Pons. Towards high-resolution large-scale multi-view stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1437, 2009.

[Wadhwa et al., 2013] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Transactions on Graphics*, 32(4):80:1–80:10, 2013.

[Wang et al., 2004] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[Wang et al., 2011] Oliver Wang, Manuel Lang, M. Frei, Alexander Hornung, Aljoscha Smolic, and Markus Gross. StereoBrush: Interactive 2D to 3D conversion using discontinuous warps. In *Proceedings of Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 47–54, 2011.

[Wanner and Goldluecke, 2012a] Sven Wanner and Bastian Goldluecke. Globally consistent depth labeling of 4D light fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48, 2012.

[Wanner and Goldluecke, 2012b] Sven Wanner and Bastian Goldluecke. Spatial and angular variational super-resolution of 4D light fields. In *Proceedings of European Conference on Computer Vision*, pages 608–621, 2012.

[Wanner et al., 2011] Sven Wanner, Janis Fehr, and Bernd Jähne. Generating EPI representations of 4D light fields with a single lens focused plenoptic camera. In *Proceedings of International Symposium on Advances in Visual Computing*, pages 90–101, 2011.

[Ward et al., 2011] Ben Ward, Sing Bing Kang, and Eric P. Bennett. Depth director: A system for adding depth to movies. *IEEE Computer Graphics and Applications*, 31(1):36–48, 2011.

[Wetzstein et al., 2011] Gordon Wetzstein, Douglas Lanman, Wolfgang Heidrich, and Ramesh Raskar. Layered 3D: Tomographic image synthesis for attenuation-based light field and high dynamic range displays. *ACM Transactions on Graphics*, 30(4):95:1–95:12, 2011.

[Wetzstein et al., 2012] Gordon Wetzstein, Douglas Lanman, Matthew Hirsch, and Ramesh Raskar. Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Transactions on Graphics*, 31(4):80:1–80:11, 2012.

[Wetzstein et al., 2013] Gordon Wetzstein, Ivo Ihrke, and Wolfgang Heidrich. On plenoptic multiplexing and reconstruction. *International Journal of Computer Vision*, 101(2):384–400, 2013.

[Wilburn et al., 2005] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio R. Antúnez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24(3):765–776, 2005.

[Wood et al., 1997] Daniel N. Wood, Adam Finkelstein, John F. Hughes, Craig E. Thayer, and David H. Salesin. Multiperspective panoramas for cel animation. In *Proceedings of ACM SIGGRAPH*, pages 243–250, 1997.

[Wood et al., 2000] Daniel N. Wood, Daniel I. Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H. Salesin, and Werner Stuetzle. Surface light fields for 3D photography. In *Proceedings of ACM SIGGRAPH*, pages 287–296, 2000.

[Woods et al., 1993] Andrew Woods, Tom Docherty, and Rolf Koch. Image distortions in stereoscopic video systems. In *Proceedings of the SPIE, Volume 1915, Stereoscopic Displays and Applications IV*, pages 36–48, 1993.

[Yang et al., 2002] Jason C. Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. A real-time distributed light field camera. In *Proceedings of Eurographics Workshop on Rendering*, pages 77–86, 2002.

[Yu and McMillan, 2004] Jingyi Yu and Leonard McMillan. General linear cameras. In *Proceedings of European Conference on Computer Vision*, pages 14–27, 2004.

[Yu et al., 2001] Yizhou Yu, Andras Ferencz, and Jitendra Malik. Extracting objects from range and radiance images. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):351–364, 2001.

[Yu et al., 2010] Jingyi Yu, Leonard McMillan, and Peter Sturm. Multi-perspective modelling, rendering and imaging. *Computer Graphics Forum*, 29(1):227–246, 2010.

[Zach et al., 2009] Christopher Zach, Marc Niethammer, and Jan-Michael Frahm. Continuous maximal flows and Wulff shapes: Application to MRFs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1911–1918, 2009.

[Zhang and Chen, 2004] Cha Zhang and Tsuhan Chen. A self-reconfigurable camera array. In *Proceedings of Eurographics Symposium on Rendering*, pages 243–254, 2004.

[Zhang and Chen, 2006] Cha Zhang and Tsuhan Chen. *Light Field Sampling*. Morgan & Claypool Publishers, 2006.

[Ziegler et al., 2007] Remo Ziegler, Simon Bucheli, Lukas Ahrenberg, Marcus Magnor, and Markus Gross. A bidirectional light field–hologram transform. *Computer Graphics Forum*, 26(3):435–446, 2007.

[Zitnick and Kang, 2007] C. Lawrence Zitnick and Sing Bing Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1):49–65, 2007.

[Zitnick et al., 2004] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608, 2004.

[Zomet et al., 2003] Assaf Zomet, Doron Feldman, Shmuel Peleg, and Daphna Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):741–754, 2003.

[Zwicker et al., 2006] Matthias Zwicker, Wojciech Matusik, Frédo Durand, and Hanspeter Pfister. Antialiasing for automultiscopic 3D displays. In *Proceedings of Eurographics Symposium on Rendering*, pages 73–82, 2006.

A

Appendix: Curriculum Vitae

Changil Kim

Contact (Office)
Disney Research Zurich
Stampfenbachstrasse 48
8006 Zurich
Switzerland

kimc@disneyresearch.com
+41 (0) 44 632 29 51
http://graphics.ethz.ch/~kimc

Contact (Home)
Schwamendingenstrasse 1
8050 Zurich
Switzerland

changilkim@gmail.com
+41 (0) 76 267 88 28

EDUCATION

Mar. 2011 – present **Doctoral Studies** *Zurich, Switzerland*
Eidgnössische Technische Hochschule Zürich (ETH Zurich), Institute of Visual Computing
- Thesis topic: 3D Reconstruction and Rendering from High Resolution Light Fields
- Advisor: Prof. Markus Gross; Co-advisor: Dr. Alexander Sorkine-Hornung (Disney Research Zurich)

Oct. 2010 **Master of Science in Computer Science** *Zurich, Switzerland*
Eidgnössische Technische Hochschule Zürich (ETH Zurich), Dept. of Computer Science
- Thesis: Scene Reconstruction from a Light Field
- Advisors: Prof. Markus Gross, Dr. Wojciech Matusik, Dr. Simon Heinzle

Feb. 2005 **Bachelor of Science in Electrical Engineering and Computer Science** *Daejeon, Korea*
Korea Advanced Institute of Science and Technology (KAIST), Dept. of EECS, Div. of CS
- Thesis: Streaming High-Definition Television over the Wired Network
- Advisors: Prof. Kilnam Chon, Prof. Chin-Wan Chung

RESEARCH AND WORK EXPERIENCES

Dec. 2010 – present **Research Assistant** *Zurich, Switzerland*
ETH Zurich and Disney Research Zurich (double affiliation)
- Working on light field rendering, image-based 3D reconstruction, and image and video processing

June 2009 – Jan. 2010 **Research Intern** *Zurich, Switzerland*
Disney Research Zurich
- Worked on perceptual studies on stereoscopy and on stereoscopic deghosting and descattering algorithms

Appendix: Curriculum Vitae

Jan. 2005 – Aug. 2008 **Manager** *Seoul, Korea*
SK Telecom Co., Ltd., Research and Development Center
- Worked on mobile TV services, especially data broadcasting security
- Developed security solutions for digital media, including conditional access system (CAS) and digital rights management system (DRM), and content distribution frameworks for consumer services over mobile phones

Aug. 2004 – Dec. 2004 **Software Engineer** *Daejeon, Korea*
KAIST
- Worked on interactive toys for children with attention deficit hyperactivity disorder (ADHD); prototyped using projector-camera system

Feb. 2000 – Mar. 2003 **Software Engineer** *Seoul, Korea*
Insung Information Co., Ltd.
- Worked on voice over IP (VoIP) and developed prototype based on H.323
- Developed computer-telephony integration, unified messaging system, and network quality measurement solution

Feb. 1999 – Dec. 1999 **System Engineer** | **Software Engineer** *Seoul, Korea*
Cyberbank Co.
- Developed PDA with CDMA wireless connection; ported firmware and operating system
- Developed image processing engine and image-based authentication system

TEACHING EXPERIENCES

Autumn 2011 – 2014 **Teaching Assistant** *Zurich, Switzerland*
Computer Graphics, ETH Zurich

Spring 2012 – 2014 **Teaching Assistant** *Zurich, Switzerland*
Informatik I, ETH Zurich

Spring 2011 **Teaching Assistant** *Zurich, Switzerland*
Design of Digital Circuits, ETH Zurich

PUBLICATIONS

Sept. 2015 Changil Kim, Kartic Subr, Kenny Mitchell, Alexander Sorkine-Hornung, Markus Gross.
"Online View Sampling for Estimating Depth from Light Fields."
Proceedings of IEEE International Conference on Image Processing (ICIP), oral presentation, among top 10% (to appear)

Dec. 2014 Changil Kim, Ulrich Müller, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, Markus Gross.
"Memory Efficient Stereoscopy from Light Fields."
Proceedings of International Conference on 3D Vision (3DV), oral presentation

July 2013 Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, Markus Gross.
"Scene Reconstruction from High Spatio-Angular Resolution Light Fields."
ACM Transactions on Graphics, 32(4); Proceedings of ACM SIGGRAPH

May 2013 Simon Wenner, Jean-Charles Bazin, Alaxander Sorkine-Hornung, Changil Kim, Markus Gross.
"Scalable Music: Automatic Music Retargeting and Synthesis."
Computer Graphics Forum, 32(2); Proceedings of Eurographics

Dec. 2011 Changil Kim, Alexander Hornung, Simon Heinzle, Wojciech Matusik, and Markus Gross.
"Multi-Perspective Stereoscopy from Light Fields."
ACM Transactions on Graphics, 30(6); Proceedings of ACM SIGGRAPH Asia

THESES

Sept. 2010 Changil Kim. "Scene Reconstruction from a Light Field." Master's Thesis, ETH Zurich

Dec. 2004 Changil Kim. "Streaming High-Definition Television over the Wired Network." Bachelor's Thesis, KAIST

HONORS AND AWARDS

2008 – 2010 **ETH Scholarship** *Zurich, Switzerland*

Apr. 2007 **IR52 Jang Young Shil Award** *Seoul, Korea*
Awarded by Ministry of Science and Technology of Korea, Korea Industrial Technology Association, and Maeil Business Newspaper

Dec. 2006 **Korea Internet Award** *Seoul, Korea*
Awarded Presidential Prize by Ministry of Information and Communication of Korea, National Internet Development Agency of Korea, and Commerce Net Korea

Nov. 2006 **New Radio Technology Award** *Seoul, Korea*
Awarded Minister of Information and Communication Prize by Korea Radio Promotion Association and Electronic Times

1996 – 2005 **Korean Government Scholarship** *Daejeon, Korea*

MOVIE CREDITS

2015 "Cinderella," Walt Disney Pictures, Visual Effects

2014 "Maleficent," Walt Disney Pictures, Visual Effects

PATENTS

2007 – present 1 US patent and 5 Korean patents granted; 3 US patent applications filed

OTHER ACTIVITIES

Feb. 2007 – Jan. 2008 **President** *Seoul, Korea*
"The Band," SK Telecom Co., Ltd.

COMPUTER SKILLS

C/C++, MATLAB, Python, Java, CUDA, OpenCL, OpenGL/GLSL, WebGL, LaTeX, HTML, JavaScript, PHP, SQL
Development environments: Vim, GNU/Linux and Unix tools, Xcode, Visual Studio

LANGUAGES

Korean (native)
English (proficient)
German (basic)

SUPERVISED STUDENTS

2013 Matan Zohar, Intern at Disney Research Zurich *Zurich, Switzerland*

2013 Guo Qi, Visiting bachelor student at ETH Zurich *Zurich, Switzerland*

2012 Werner Randelshofer, Master student at ETH Zurich *Zurich, Switzerland*

2012 Ulrich Müller, Master student at ETH Zurich *Zurich, Switzerland*

2012 Christian Reiter, Bachelor student at ETH Zurich *Zurich, Switzerland*

ACADEMIC SERVICE

Reviewer for:
ACM SIGGRAPH
ACM SIGGRAPH Asia
Eurographics
IEEE Transactions on Visualization and Computer Graphics
Computer Graphics Forum
Computers & Graphics
Journal of Electronic Imaging
International Conference on 3D Vision
European Conference on Computer Vision
High Performance Graphics

REFERENCES

Prof. Markus Gross (PhD advisor)
Professor, ETH Zurich | Director, Disney Research Zurich
Universitätstrasse 6, 8092 Zurich, Switzerland
grossm@inf.ethz.ch

Dr. Alexander Sorkine-Hornung (PhD advisor)
Senior Research Scientist, Disney Research Zurich
Stampfenbachstrasse 48, 8006 Zurich, Switzerland
alexander@disneyresearch.com

Prof. Wojciech Matusik (MSc advisor)
Associate Professor, MIT
32 Vassar Street, Cambridge, MA 02139, USA
wojciech@mit.edu

Dr. Simon Heinzle (MSc advisor)
CEO, Gimalon AG
Technoparkstrasse 1, 8005 Zurich, Switzerland
simon@gimalon.com

Prof. Kilnam Chon (BSc advisor)
Professor, Keio University | Professor Emeritus, KAIST
Fujisawa, Kanagawa 252-8520, Japan
chon@cosmos.kaist.ac.kr

Prof. Chin-Wan Chung (BSc advisor)
Professor, KAIST
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
chungcw@cs.kaist.ac.kr

Zurich, August 2015