Diss.-No. ETH 21510

Acquisition, Processing and Display for 3D Live-Action Cinema and Television

A dissertation submitted to
ETH Zurich

for the Degree of
Doctor of Sciences

presented by
Jeroen van Baar
M.Sc. Delft University of Technology, the Netherlands
born 04 September 1973
citizen of the Netherlands

accepted on the recommendation of
Prof. Dr. Markus Gross, examiner
Prof. Dr. Marc Pollefeys, co-examiner
Dr. Paul Beardsley, co-examiner

2013

Abstract

Three-dimensional cinema and television involves the presentation of a separate image to a viewer's left and right eyes, in order to invoke a depth perception. Three-dimensional cinema and television provides filmmakers with an additional cue to aid in their storytelling. Current acquisition and manipulation approaches make it difficult to effectively exploit the additional depth dimension. In this thesis we examine the pipeline of acquisition, processing and display, and propose methods and approaches which make it easier to exploit the depth dimension, while also aiming to improve the quality of the three-dimensional viewing experience.

Computing a depth value for each pixel in the video images of a captured scene is a difficult task. We propose an acquisition system where a central, high quality film camera is supported with additional satellite sensors. Rather than using sensors of a single modality, e.g. visible light cameras, we propose to use additional modalities. Besides lower quality visible light cameras, we also incorporate a Time-of-Flight depth camera and a thermal camera. By combining sensors of different modalities we aim to provide more information for computing per-pixel depth. The satellite cameras allow for better occlusion reasoning of the scene. A depth camera provides a direct measure of scene depth, albeit at a low resolution. Finally, a thermal imaging camera provides information to correctly discern between different scene elements, when those scene elements are imaged as regions with similar colors. We propose a method to combine the information from multiple modalities and demonstrate that we can compute high quality depth maps.

Since we are dealing with motion pictures, it's not sufficient to compute depth only for a single instant in time. The computed depth should be temporally consistent for the video. We argue that the temporally consistent depth is of most importance for foreground objects in a scene. We propose an interactive approach which propagates segmented foreground objects from a begin and end frame of a shot, to the frames in between. By grouping pixels with similar photometric and thermal properties into so-called superpixels, we reduce the complexity from per-pixel to per-superpixel. We then pose the problem as a labeling problem for superpixels over time, where the label that is assigned to each superpixel indicates to which segment that superpixel belongs. We show that this information can be directly exploited in the depth computation, where the segments are used as prior knowledge in that computation.

For three-dimensional acquisition using a stereo pair of cameras, the arrangement of the cameras at acquisition time determines the amount of depth that can be perceived by the viewers. The depth of the underlying scene has to be recovered in order to change the amount of depth perceived. In general, the processing of three-dimensional content cannot be performed independently for the left and right eye images, but should also take the underlying scene depth into account. We propose a processing method which can copy elements from a scene captured with one particular arrangement of cameras, and then paste those elements into a scene with a different arrangement. We demonstrate a method that even in the case where the recovered depth cannot be accurately estimated, we can robustly copy and paste elements. We further demonstrate how the underlying scene depth can be exploited when the element is pasted, and avoid the difficult task of scene in-painting, while aiming to conform to the stereo properties of the target scene.

Display of stereoscopic three dimensional content provides the user the ability to perceive a depth impression. The key factor is to ensure that each eye is only stimulated with the corresponding image of the stereoscopic image pair. The case where information intended for one eye is perceived by the other eye, is denoted as crosstalk or ghosting. The presence of ghosting may result in objects being perceived at the incorrect depth, and even result in the depth impression being entirely lost. Ghosting also puts a relatively heavy burden on the visual system of the viewer, with visual fatigue as the consequence. We identify that display systems are not perfect and propose a computational approach to mitigate the occurrence of ghosting in stereoscopic three dimensional display systems. Our approach is based on incorporating perceptual metrics to compensate the input images in such a way as to provide a perceptually more optimal viewing experience.

Zusammenfassung

In dieser Doktorarbeit befassen wir die Pipeline zur Erfassung, Verarbeitung und Darstellung von dreidimensionalen Inhalten für Kino und Fernsehen.

Die Beiträge die in dieser Doktorarbeit gemacht werden, können als wie folgt zusammengefasst werden:

- **Verfassungssystem auf einem einzigen Referenzkamera, durch multimodale Satellitsensoren unterstützt, für die Berechnung der Tiefekarten und Segmentierung**
- **Fusion multimodaler Sensorinformationen zu berechnen der Tiefekarten mit einem lokalen Verfahren.**
- **Interaktive Video Segmentierung Methode mit multimodalen Sensorinformationen. Das Ergebnis der Segmentierung wird zur Berechnung verbesserte Tiefekarten verwendet.**
- **Ein System für Kopieren und Einfügung Bearbeitung von stereoskopischen 3D-Inhalten mit Tiefe und Segmentierung Informationen.**
- **Ein Wahrnehmungsbasiertes System für die Kompensation der Lichtverschmutzung durch sogenannte Geisterbilder in stereoskopische 3D Abbildungssysteme. Das Kompensationssystem ist allgemein und kann für alle Formen der additiven Lichtverschmutzung in Abbildungssysteme angewendet werden.**

Cinematographers und Kamerabetreiber haben sich gewöhnt um mit einer einzigen Kamera zu erfassen. Wir schlagen daher eine multimodalen Verfassungssystem vor, mit einer zentralen hochqualitativen Kamera der mit verschiedenen Arten von Sensoren erweitert wird, um die Berechnung der Tiefekarten und Segmentierung zu unterstützen. Unser Prototyp zeigt, dass es relativ einfach ist, ein solches System zu bauen. Auch der Durchführung von geometrischen Kalibrierung und Farbkalibrierung ist relativ einfach.

Wir beschreiben eine lokale Methode basiert auf der Fusion der verschiedenen Modalitäten für die Berechnung von Tiefekarten. Tiefekarten werden für die hohe Qualität Referenz Kamera in das Verfassungssystem berechnet. Experimentelle Ergebnisse werden für Szenen mit dynamischen Objekten und Hintergrundkram gezeigt. Okkludierungen, Texturlose Regionen, wiederholende Texturen, oder ähnlich farbigen Vorder- und Hintergrundobjekte können vor Probleme sorgen in Methoden die sich nur auf

Farbkonsistenz verlassen. Mehrere SatellitKameras ermöglichen es uns Okklusionsregionen besser abzuschatzen, durch der Farbkonsistenz zwischen der Referenzkamera und die SatellitKameras auf der linken Seite, und die Farbkonsistenz zwischen der Referenzkamera und die SatellitKameras auf der rechten Seite, zu vergleichen. Die Fusion von Stereo mit Time-of-Flight Tiefedaten ergibt der richtigen Rekonstruktion von Texturlose Regionen wie Hintergrundwänden. Daneben können gleichfarbige, aber in unterschiedlichen Tiefen überdeckende Flächen, korrekt rekonstruiert werden. Von besonderem Interesse ist der Fall, wenn menschliche Subjekte oder Körperteile sich überdecken. Wir haben gezeigt, dass verschiedene Subjekte unterschiedliche thermische Signaturen haben können. Daher, durch die Fusion von thermischen Daten kann ein okkludierende Kontur gefunden werden, obwohl die Hautfarbe ähnlich ist. Wir vergleichen die Fälle von Fusion von Stereo mit dem Time-of-Flight Tiefedaten, Fusion mit den thermischen Daten und Fusion mit den beiden Time-of-Flight Tiefe und thermischen Daten. Obwohl jede dieser Modalitäten separat zu ein verbesserte Tiefekarte beitragen können, die Kombination aus beiden gibt Tiefekarte mit den beste Ergebnisse.

Eine zentrale Aufforderung bei der Berechnung der Tiefekarten ist die Schätzung der Okklusionsbereichen in einer Szene. Die Verwendung mehrerer Satellitkameras auf beiden Seiten einer Referenzkamera, verhilft zu einer besseren Schätzung. Um Kosten und Stellfläche der Verfassungssystem zu reduzieren, schlagen wir vor, geringere Qualität Satellitkameras zu verwenden. Dagegen hatten geringere Qualität Kameras mehr Bildrauschen als die hochqualitativen Kamera. Dies gilt vor allem in dunkleren Bildbereichen. Darüber hinaus haben die Satelliten und Referenzkameras auch sehr unterschiedliche Farbräume. Diese Eigenschaften beeinflussen die Genauigkeit der Farbkonsistenz zwischen den Satellitkameras und dem Referenzkamera, und damit die gesamte Präzision. Das berechnen von Tiefekarten basiert auf Farbkonstanz allein, werd immer von Vieldeutigkeiten leiden. Fusion mit zusätzlichen Modalitäten ist daher eine vielversprechende Richtung zur Lösung einiger dieser Vieldeutigkeiten. Die Auflösung des Time-of-Flight Kameras ist sehr niedrig im Vergleich zu den Referenzkamera. Feine Details, wie die Blätter einer Pflanze, können daher nicht genau durch der Time-of-Flight Kamera erfasst werden. Fusion mit niedriger Auflösung Time-of-Flight Tiefe funktioniert deshalb am besten für Bereiche ohne feine Details. Wärmebilder sind besonders nützlich, wenn das thermische Kontrast ausreichend hoch ist. Dies ist typischerweise der Fall bei Szenen mit menschlichen Akteuren.

Wir beschreiben ein interaktives Video Segmentierung Methode für die Segmentierung mehrere Vordergrund Objekte vom Hintergrund. Unsere Meth-

ode propagiert bekannte Segmentierungen für das erste und letzte Bild zu der Zwischenbilder in einer Videosequenz. Die propagierung stützt sich auf der Übereinstimmung von Superpixeln in der Videosequenz, ohne Annahme auf die Art der Bewegungen in einer Szene. Unsere Methode kann deshalb bewegten Kameras und nicht-starr bewegten Objekten verarbeiten. Das Ausnutzen mehreren Modalitäten trägt zu ein mehr robuster Übereinstimmung der Superpixel zwischen Frames einer Sequenz bei. Die Propagierung von bekannte Segmentierungen kann okkludierende Objekte verarbeiten. In sofern dass Objekte im Vordergrund in einer Sequenz sowohl in der ersten und letzten Rahmens sind, können sie dann für der Zwischenbilder verschwinden und wieder erscheinen. Falls optical flow Informationen verfügbar sind, kann es einfach für die Übereinstimmung von Superpixel eingearbeitet werden.

Ein vollautomatisches System kann zu ein falschen Segmentierung führen. Wir schlagen daher vor, um einen Benutzer interaktiv die Propagierung einer Segmentierung zu begleiten. Wir benötigen Korrekturen nur auf einen groben Niveau statt auf Pixelniveau. Das verringert die Belastung für den Benutzer. Genaue Segmentgrenzen werden dann in einem nachfolgenden Schritt produziert. Mehrere Modalitäten können dann genutzt werden zur Lösung der Farbenvieldeutigkeiten, und produzieren dann besseren Segmentgrenzen. Die Segmentgrenzen sind zeitlich stabil, weil sie genau die Grenzen der Objekte im Video passen. Wir können die Grenzen als Randbedingungen in die Berechnung der Tiefekarten verwenden, so dass die Tiefesilhouetten zeitlich mehr stabil werden.

Wir beschreiben ein System für 3D Kopieren & Einfügen. Das System baut das 2D Kopieren & Einfügen für Standbilder zu stereoskopischen 3D aus. Die Rekonstruktion der Tiefekarte für die Szene ist die grundlegende Operation in diesem System. Die rekonstruierten Tiefekarten können bei der Durchführung des interaktiven Segmentierung benutzt werden, für die Propagierung des Segmentierungsergebnis für das Bild entsprechend mit einem Auge, zu das Bild entsprechend mit dem anderen Auge. Die rekonstruierten Tiefekarten können auch bei die Zusammensetzung der segmentierten Objekte in der Zielszenen benutzt werden. Segmentierung, die Propagierung, und Zusammensetzung werden alle von höherer Qualität Tiefkarten profitieren. Direktes Zusammensetzung mit benutzung eine Tiefekarte, würde eine fehlerfreie Tiefekarte benötigen. Fehlerfreie Tiefekarten können jedoch selten für allgemeine Szenen erhalten werden. Zusammensetzung basiert auf Tiefekarten mit Fehlern, können stattdessen Proxygeometrie und parametrische Verzögungen benutzen.

Bei Zusammensetzung unter verschiedenen Orientierungen, oder in ein Ziel-

szene mit verschiedenartige stereo Parameter, könnte Okklusionsbereichen wieder sichtbar werden. Für realistische Ergebnisse müssten diese wieder sichtbare Okklusionsbereichen eingemalt werden. Wir zeigen stattdessen dass, durch die Anwendung der entsprechenden Einschränkungen für das berechnen der parametrische Verzögungen, wieder sichtbare Okklusionsbereichen ganz vermieden werden können. Die Ergebnisse bleiben jedoch immer noch überzeugend.

Wir beschreiben ein System für die Kompensation von Geisterbildern und Lichtstreuung. Durch die Formulierung der Kompensation als Optimierungsproblem, können wir den System in alle Fälle von zusatzliches Lichtverschmutzung anwenden. Als solches ist unsere Formulierung eine Verallgemeinerung der vorhandenen subtraktiven Kompensationsverfahren. Da wir für den menschlichen Beobachter kompensieren werden, sollten wir die Eigenschaften des menschlichen visuellen System nutzen. Wir zeigen wie Beobachtungsmetriken in die Optimierungsformulierung eingearbeitet werden können. Insbesondere durch die Integrierung der Kontrastempfindlichkeitsfunktion und das Lösen des resultierenden Optimierungsproblem, wird der restliche Fehler in Regionen, in denen das menschliche visuelle System weniger empfindlich ist, verbreitet. Am wichtigsten ist, dass die Wahrnehmbarkeit von möglicherweise widersprüchliche Randcues für Stereosehen, für Wahrnehmungsbasierte Deghosting reduziert ist. Dies macht gerade das beobachten von stereoskopische 3D Abbildungssysteme noch komfortabler. Ein Benutzerstudie wurde durchgeführt, um sicherzustellen, dass unsere Methode in der Tat von Benutzer bevorzugt wird, statt einfache subtraktive Kompensation.

Summary

In this thesis we address the pipeline for acquisition, processing and display of three-dimensional contents for cinema and television.

The contributions made in this thesis can be summarized as:

- **Acquisition system based on a single reference camera, supported by multi-modal satellite sensors, for computing depth maps and segmentation**
- **Fusion of multi-modal sensor information to compute depth maps using a local method.**
- **Interactive video segmentation approach using multi-modal sensor information. The result of the segmentation is used for computing improved depth maps.**
- **A framework for copy and paste editing of stereoscopic 3D content using depth and segmentation information.**
- **A perceptually-based framework for the compensation of light pollution due to ghosting in stereoscopic 3D displays. The framework is general and can be applied to all forms of additive light pollution in display systems.**

Cinematographers and camera operators are used to capture with a single camera. We therefore propose a multi-modal capture system, using a central high quality reference camera augmented with different types of sensors to support the computation of depth maps and segmentation. Our prototype system demonstrates that it is relatively straightforward to build such a system, including performing geometric calibration and color calibration.

We describe a local method based on fusion of the different modalities for computing depth maps. Depth maps are computed for the high quality reference camera in the capture system. Experimental results are shown for scenes with dynamic objects and background clutter. Occlusions, textureless regions, repeated textures, or similarly colored fore- and background objects may pose problems in methods that rely only on color consistency. Multiple satellite cameras allow us to better estimate occlusion regions by comparing the color consistency between the reference camera and the satellite cameras on the left side, to the color consistency between the reference camera and the satellite cameras on the right side. The fusion of stereo with Time-of-Flight

depth data results in the correct reconstruction of textureless regions such as background walls. In addition, surfaces of the same color, but overlapping at different depths can be correctly reconstructed. Of particular interest is the case where human subjects or body parts are overlapping. We showed that different subjects may have different thermal signatures. Therefore, by also fusing the thermal data, an occluding contour can be found even though the skin color is similar. We compared the cases of fusion of stereo with only the Time-of-Flight depth data, fusion with only the thermal data, and fusion with both Time-of-Flight depth and thermal data. Although each of these modalities separately can help improve the depth map, the combination of both gives the best result.

A key challenge in computing depth maps is the estimation of occlusion areas in a scene. Using multiple satellite cameras on either side of a reference camera, helps to better estimate occlusions. To reduce cost and physical footprint of the acquisition system, we propose to use lower quality satellite cameras. However, lower quality cameras exhibit more noise than the high quality reference camera. This is particularly true in low light areas. In addition, the satellite and reference cameras also have very different color spaces. These properties affect the accuracy of the color consistency between the satellite cameras and the reference camera, and therefore the overall accuracy as well. Computing depth based on color consistency alone will always suffer from ambiguities. Fusion with additional modalities is therefore a promising direction to help solve some of these ambiguities. The Time-of-Flight depth camera resolution is very low compared to the reference camera. Fine details, such as the leaves of a plant, are therefore not accurately captured with the Time-of-Flight depth camera. Fusion with low resolution Time-of-Flight depth thus works best for areas without fine details. Thermal images are most useful when thermal contrast is sufficiently high. This is typically the case for scenes with human actors.

We describe an interactive video segmentation approach to segment multiple foreground objects from the background. Our approach propagates known segmentations for the first and last frame to the intermediate frames in a video sequence. The propagation relies on the matching of superpixels across the video sequence, without any assumption on the motions in a scene. Our method can thus handle moving cameras and non-rigidly moving objects. Exploiting multiple modalities helps to make the matching of superpixels between frames of a sequence more robust. The propagation of known segmentations can handle occluding objects. Provided that foreground objects within a sequence are present in both the first and last frame, they may then disappear and re-appear for the intermediate frames. If optical flow information is available, it can be easily incorporated for the matching of superpixels.

A fully automated method may produce the wrong segmentation. We thus propose to employ a user to interactively guide the propagation of a segmentation labeling. We require corrections only at a coarse level, rather than at the pixel level, which reduces the burden on the user. Accurate segment boundaries are produced in a subsequent refinement step. Multiple modalities can then be exploited to help resolve color ambiguities and result in better refinement boundaries. The segment boundaries are temporally stable as they accurately match the object boundaries in the video. We can use the boundaries as constraints in the computation of depth maps, so that the depth silhouettes become temporally more stable as well.

We describe an end-to-end system for 3D copy & paste, which extends 2D copy & paste for still images to stereoscopic 3D. The reconstruction of the depth map for the scene is the fundamental operation in this system. The reconstructed depth maps can be used when performing the interactive segmentation, for the propagation of the segmentation result for one eye image to the other eye image, and for composition of the segmented objects into the target scenes. Segmentation, propagation, and composition will all benefit from higher quality depth maps. Direct composition based on the depth map on the other hand, would require an error-free depth map. Error-free depth maps are rarely obtained for general scenes however. Compositing based on depth maps with errors can instead be done using proxy geometry and parametric warps.

When compositing under different orientation, or into a target scene with different stereo parameters, disocclusions could occur. For realistic results these disocclusions would have to be inpainted. Instead we show that by applying the appropriate constraints to compute the parametric warps, disocclusions can be avoided altogether, while still achieving compelling results.

We describe a framework for the compensation of ghosting and scattering. By formulating the compensation as an optimization problem, we can apply the framework to additive light pollution in general. As such, our formulation is a generalization of existing subtractive compensation methods. Since we are compensating for human observers, we should exploit the properties of the human visual system. We show how we can incorporate perceptually-based metrics into the optimization formulation. Specifically, by incorporating the Contrast Sensitivity Function and solving the resulting optimization problem, the residual error is distributed to regions where the human visual system is less sensitive to them. Most importantly, the perceptibility of possibly conflicting edge cues for stereopsis is reduced for perceptual-based deghosting. This makes watching stereoscopic 3D displays more comfortable. A user study was conducted to verify that our perceptually-based compensa-

tion method is indeed generally preferred over straightforward subtractive compensation.

Acknowledgements

I am greatly indebted to the many people who have, and continue to have supported and inspired me: Joe Marks, Markus Gross, Marc Pollefeys, Paul Beardsley, Hanspeter Pfister, Wojciech Matusik, Matthias Zwicker, the researchers, post-docs and students at ETH and Disney Research Zurich, for their many inspiring research projects and helpful discussions. The co-authors on various papers: Steven Poulakos, Wojciech Jarosz, Derek Nowrouzezahrai, Rasmus Tamstorf, Wan-Yen Lo, Anselm Grundhoefer, Max Grosse, Mark Mine, Bei Yang, David Rose. The colleagues at the Walt Disney Company. My apologies to anyone I have forgotten to mention.

To my parents: thank you for all you've done.

Last, but certainly not least, my wife and children: thank you for your love and support.

Contents

List of Figures xvii

List of Tables xxi

Notation xxiii

1 Introduction 1
- 1.1 Motivation . . . 3
- 1.2 Contributions . . . 6
- 1.3 Organization . . . 8
- 1.4 Publications . . . 8

2 Related Work 11
- 2.1 Acquisition . . . 11
- 2.2 Processing—Depth Maps . . . 13
- 2.3 Processing—Segmentation and Depth Maps . . . 14
- 2.4 Stereoscopic Editing—Stereo Copy & Paste . . . 16
- 2.5 Stereoscopic Display—Deghosting . . . 18

3 Multi-Modal System for Acquisition 21
- 3.1 Introduction and Motivation . . . 22

3.2 System . . . 23
3.2.1 High Quality Reference Camera . . . 23
3.2.2 Satellite Cameras . . . 24
3.2.3 Time-of-Flight Depth Camera . . . 25
3.2.4 Far Infrared Thermal Camera . . . 25
3.2.5 Sensors Positioning . . . 26
3.2.6 Synchronization . . . 27
3.3 Multi-Sensor Multi-Modal Acquisition . . . 27
3.3.1 Expected Benefit of Satellite Cameras / Sensors . . . 27
3.3.2 Expected Benefit of Adding ToF Depth Sensor . . . 28
3.3.3 Expected Benefit of Adding Thermal Sensor . . . 28
3.4 Calibration . . . 28
3.4.1 Geometric Calibration . . . 28
3.4.2 Photometric Calibration . . . 31
3.5 Discussion . . . 32

4 Processing—Depth Maps 33
4.1 Motivation . . . 34
4.2 Problem Formulation . . . 35
4.3 Sensor Fusion via Local Method . . . 36
4.3.1 Multi-View Stereo Depth using Satellite Cameras . . . 38
4.3.2 Fusion with Time-of-Flight Depth . . . 40
4.3.3 Reprojection onto Satellite Cameras, with Occlusion Reasoning . . . 41
4.3.4 Winner Take All . . . 44
4.4 Plane Fitting for Improving Depth Estimates . . . 45
4.5 Smoothing . . . 47
4.6 Results for Fusion via Local Method . . . 48
4.7 Fusion with Thermal Signal . . . 50
4.8 Results . . . 52
4.8.1 Comparison of Modalities . . . 53
4.9 Discussion . . . 53
4.9.1 Temporal Consistency . . . 54

5 Processing—Segmentation 59
5.1 Motivation . . . 60
5.2 Interactive Segmentation . . . 62
5.2.1 Segmentation Propagation as Energy Minimization . . . 63
5.2.2 Matching Superpixels . . . 64
5.2.3 Incorporating Optical Flow . . . 67
5.2.4 Initial Propagation . . . 68
5.2.5 Incorporating Temporal Information for Propagation . . . 68

5.2.6 Interactive Segmentation Correction 71
5.3 Segmentation Boundary Refinement 71
5.4 Exploiting Multiple Modalities 72
5.5 Results 72
5.6 Application: Depth Maps 74
5.6.1 Simplified Belief Propagation 74
5.6.2 Segment Boundaries for Message Passing 77
5.7 Results 77
5.8 Discussion 78

6 Processing—Stereoscopic Editing 81
6.1 Introduction 82
6.2 Stereoscopic Copy & Paste 83
6.2.1 Depth Reconstruction 85
6.2.2 Selection 86
6.2.3 Composition 89
6.3 Results 97
6.4 Discussion 99

7 Stereoscopic 3D Display 103
7.1 Motivation 104
7.2 Ghosting 106
7.3 Perceptually-based Compensation 107
7.3.1 Compensation Formulated as Optimization Problem . 107
7.3.2 Linear Perceptual Weighting 109
7.4 Perceptually-based Deghosting and Descattering 112
7.4.1 Deghosting 112
7.4.2 Descattering 113
7.5 Results 114
7.5.1 User Evaluation 118
7.6 Non-Linear Perceptual Weighting 119
7.7 Discussion 121

8 Conclusions 125
8.1 Discussion 125
8.2 Future Work 128

A Depth Maps using Active Illumination and Multi-Spectral Cameras 131
A.1 Motivation 131
A.2 Acquisition 132
A.2.1 Calibration 134

A.3 Stereo from Multi-Spectral Camera Pair 134
A.3.1 Interpolating Invalid Disparities 136
A.3.2 Extensions 138
A.4 Comparative Analysis 139
A.5 Discussion 141

Bibliography **143**

List of Figures

1.1 Wheatstone Stereoscope. 2
1.2 3D Cinema to Mobile 3D 3

2.1 Multi-camera Rig. 12

3.1 Compact Cinema Cameras. 22
3.2 Experimental System. 24
3.3 Example Dataset. 26
3.4 Thermal Camera Calibration. 30
3.5 Thermal Camera Calibration Result. 31
3.6 Photometric Calibration. 32

4.1 Overview of Depth Computation System 37
4.2 Discrete Depth Layers. 39
4.3 Time-of-Flight Camera Images. 41
4.4 Occlusion Cases. 43
4.5 Sensor Fusion using Local Method. 45
4.6 Sensor Fusion using Local Method—Examples I. 49
4.7 Sensor Fusion using Local Method—Examples II. 50
4.8 Segmentation Exploiting Thermal Signal. 51
4.9 Thermal Segmentation Prior 52
4.10 Fusion Comparisons—I 56

4.11 Fusion Comparisons—II 57

5.1 Overview of System for Segmentation Propagation and Depth. 60
5.2 Segmentation Propagation via Superpixels Matching. 63
5.3 Superpixels. 64
5.4 Matching Superpixels within Search Radius. 66
5.5 Superpixels Sequence. 69
5.6 Flowergarden Segmentation 72
5.7 Occluding Objects Segmentation 73
5.8 Before and After Boundary Refinement 75
5.9 Message Passing with Segment Boundaries 78
5.10 Segment Boundaries for Depth Maps 79

6.1 System Overview for 3D Copy & Paste 84
6.2 Workflow for 3D Copy & Paste. 84
6.3 Object Selection 86
6.4 Local Surface Orientation Alignment. 90
6.5 Rotation Constraints. 91
6.6 Stereo Billboards. 93
6.7 Stereo Billboards Compositing. 95
6.8 Shadow Synthesis. 96
6.9 Composition. 97
6.10 Stereo Copy & Paste Example Scenes. 98
6.11 Limitations. 100
6.12 Source Images for Copy & Paste. 102

7.1 Ghosting 105
7.2 Perceptually-based Compensation. 106
7.3 Perceptually-based Compensation Dataflow 108
7.4 1D and 2D Constrast Sensitivity Function 110
7.5 Perceptually-based Compensation 111
7.6 Experimental Stereoscopic Display System 114
7.7 Deghosting Comparison 116
7.8 Additional Deghosting Comparisons 116
7.9 Descattering 117
7.10 Deghosting User Evaluation 118
7.11 Ghosting Prediction Map 120
7.12 Ghosting at Sweet-spot vs. Peripheral Locations 123

8.1 Production-Ready Capture System. 129

A.1 Acquisition for Validation. 133
A.2 Multi-Spectral Acquisition 134

A.3 Multi-spectral Sensor Alignment. 135
A.4 Left-Right/Right-Left Consistency Check. 136
A.5 Occlusion Classification. 137
A.6 Disparity Interpolation. 138
A.7 Infrared Only vs. Fusion of Visible and Infrared. 139
A.8 Depth Map Validation. 140

List of Tables

5.1 Propagation of Known Segmentations. 62

7.1 Perceptual Compensation Performance 121

Notation

Here we briefly describe the notation convention we use in this thesis. Scalar variables will be written as lowercase letters, e.g., x. Vectors will be denoted as lowercase bold letters with their components as lowercase italic letters, e.g., $\mathbf{x} = (a, b, c)^T$. Vectors are assumed to be column vectors, and a row vector is then denoted as $\mathbf{x}^T$. Matrices will be denoted as uppercase bold letters, e.g., $\mathbf{M}$. A column j of the matrix is denoted as $\mathbf{m}^j$, whereas row i is denoted as $\mathbf{m}^{iT}$. A component of a matrix will be written as $m_{i,j}$, representing the value at row i and column j. The notation $\mathbf{M}_{k\times l}$ represents the $k\times l$ submatrix of $\mathbf{M}$. Finally, $diag(\mathbf{a})$ represents a diagonal matrix with the diagonal given by vector $\mathbf{a}$.

When we present equations related to projective geometry, a vector $\mathbf{x} = (x, y)^T$ will represent a 2D point. We do not explicitly distinguish between homogeneous points $(x, y, z)^T$ and inhomogeneous points $(x, y)^T$, but rather implicitly assume $(x, y, 1)^T$ in the latter case. We will sometimes distinguish points in 3D space as $\mathbf{X} = (X, Y, Z)^T$ or $(X, Y, Z, 1)^T$. In other words, a 3D point is sometimes represented as a matrix rather than a vector. With $\cong$ we denote equality up to scale, for example projection of a 3D point onto the 2D image plane will be written $\mathbf{x} \cong \mathbf{MX}$.

In all other situations we will explain what the variables and notation represent in a given equation.

CHAPTER 1

Introduction

In 1838 Wheatstone [Wheatstone, 1838] described the phenomenon of stereopsis: the observation of a scene under slightly different views as generated by an observers' left and right eye, resulting in the perception of depth. Based on this concept, Wheatstone introduced the stereoscope (see Figure 1.1) for viewing images with depth. The stereoscope presents an observer two images: one intended only to be viewed by the left eye, and one intended only to be viewed by the right eye. Since this concept mimics the human visual system, the illusion of depth can be created for an observer. The left and right eye image pair is then referred to as a stereoscopic pair of images, and the depth perception as *stereoscopic 3D*. Stereoscopic 3D therefore involves the simultaneous, or near simultaneous, display of a stereoscopic pair of images.

With the invention of motion pictures, in 1890 W.Friese-Green extended the concept of the stereoscope to introduce stereoscopic 3D motion imagery, by simultaneously displaying separate films for the left and right eye [Wikipedia–3D Film, 2013]. Although stereoscopic 3D motion pictures have been reintroduced several times since then, technological limitations for both the recording and display systems have prevented full adoption of the technology. Color differences between the left and right eye image, or synchronization problems resulted in uncomfortable viewing conditions. However, with the most recent reintroduction around 2003, technological ad-

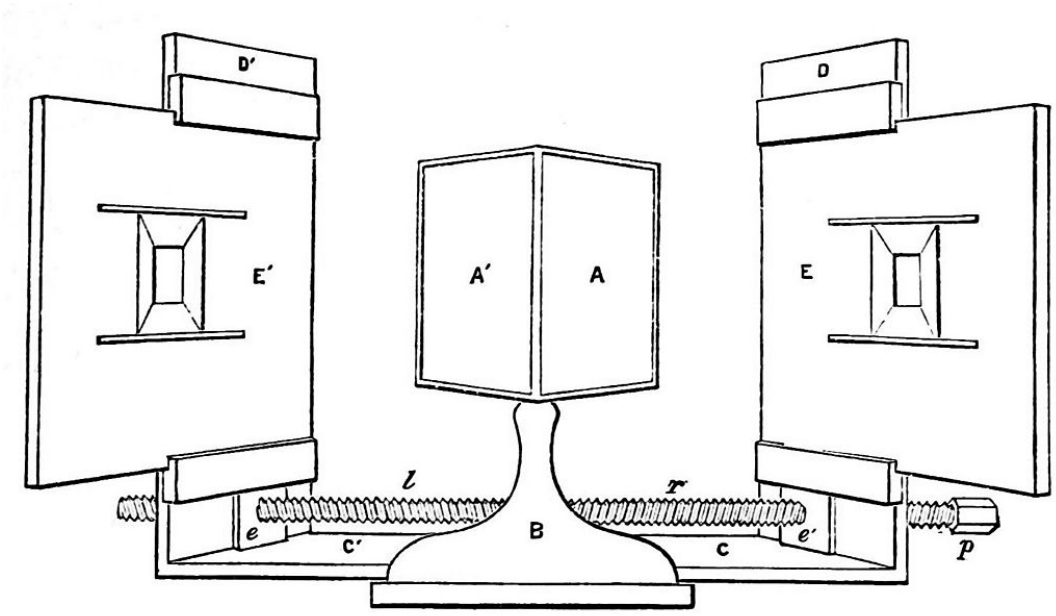


Figure 1.1: Wheatstone Stereoscope. *The stereoscope introduced by Wheatstone [Wheatstone, 1838]. The device served to demonstrate that people can perceive depth by observing a stereoscopic pair of images: one image intended to be viewed only with the left eye, and another only with the right eye. Stereoscopic 3D motion pictures were introduced based on its principles.*

vances have helped to overcome earlier limitations. Furthermore, in addition to viewing stereoscopic 3D content in the cinema, many consumer display devices can now display stereoscopic 3D content at home. Recently mobile devices with stereoscopic 3D capability also became available.

Despite the technological advances, acquisition fundamentally still occurs with two cameras, one for each eye, and more or less directly displaying the output of each. Especially in the case of motion pictures, one might want to edit the 3D content first. However, editing 3D content is not simply a matter of repeating a 2D editing operation on the image for each eye. Since the stereoscopic pair of images is intended to generate the perception of depth, when editing 3D content the depth of the underlying scene has to be taken into account. One therefore needs to be able to recover or reconstruct that depth.

Furthermore, in contrast to observing the real-world, stereoscopic 3D images are displayed on the image plane of the display device. An observer therefore has to maintain focus on that image plane, while rotating the eyes to obtain



Figure 1.2: 3D Cinema to Mobile 3D. *Stereoscopic 3D is no longer exclusive to 3D cinemas.*
Images courtesy of Nintendo ©, http://www.benjaminbernarddigital.com

a proper depth perception. This is referred to as the decoupling between vergence and accommodation [Hoffman et al., 2008, Lambooij et al., 2009]. Since vergence and accommodation decoupling is not natural, it could lead to increased cognitive load and eye strain, making the viewing experience uncomfortable. It is therefore important to ensure both high quality content and display for stereoscopic 3D.

Presenting high quality stereoscopic 3D content impacts the entire pipeline: from acquisition, to processing, to display. In this thesis we propose novel systems and methods which aim to improve acquisition, depth map computation, stereoscopic editing, and stereoscopic display.

1.1 Motivation

Computing depth maps from two or more cameras is a well-studied problem. The Middlebury Stereo evaluation [Scharstein and Szeliski, 2012] provides an overview of the many different stereo methods that have been proposed. The results are presented for a set of somewhat artificial example scenes, however computing good quality depth maps for general scenes remains a challenging

problem. The challenge in computing depth from multiple views is to determine which pixels between different cameras correspond to each other. This is achieved by examining their color similarities, and the hypothesis is that pixels with a high degree of color similarity are likely to correspond to each other. Uniformly colored objects, objects with repeating textures, or similarly colored fore- and background objects could thus result in matching ambiguities. In addition, the occurrence of occlusions may lead to additional matching ambiguities. Occlusions may arise when objects at different spatial locations are imaged from different camera locations. As a result, surfaces visible in one camera, may be obscured in another camera.

To acquire images of the scene with multiple cameras at the same resolution and color, homogeneous camera setups are used, i.e., all cameras are of same make and model. It may not be feasible to use multiple of them, e.g. due to cost or form factor of the camera. In the latter case using multiple cameras would result in a bulky setup. Smaller form factor, lower quality and hence lower cost cameras are becoming ubiquitous. Furthermore, sensors that acquire different modalities are also becoming more ubiquitous. That leads to the idea to instead combine different sensors, and different modalities with a high quality camera. The different modalities are then to be fused in order to compute depth.

The research questions we address are:

- Which kind of sensors should be included, how many and where placed?
- How should we fuse the information from multiple modalities to compute depth maps?
- Can lower quality sensors, but different modalities, be fused with high quality reference images to obtain high quality results?
- What is the contribution of each modality for computing depth maps?

An important step in video processing is segmentation of the scene into fore- and background objects or regions. Segmentation and depth maps are correlated in that segmentation boundaries typically coincide with depth discontinuities. Segmentation is also a well studied problem with many proposed solutions, however segmentation remains a challenging problem, especially for video sequences.

As with depth computation, segmentation relies on color differences between objects and background. Additional modalities could help make video segmentation more robust. Many automatic methods for video segmentation are based on the displacement of corresponding pixels between frames of a

video, often referred to as optical flow. Optical flow does not always yield reliable results, especially in the presence of large motions. For video data that is processed offline, for example in cinematic applications, user interaction would be acceptable. Provided that the amount of required interaction is limited, additional (iterative) user input could then help recover from erroneous segmentations.

Computing depth maps on each frame independently could result in temporally noisy depth maps. Often it is most important for foreground objects to maintain temporal consistency along the depth silhouettes. A known segmentation can be exploited for computing temporally more consistent depth maps.

The research questions we address are:

- How can we formulate the problem of video segmentation?
- How can we exploit multiple modality input data for segmentation?
- How can we best incorporate user input to ensure correctness of the segmentation?
- How can we achieve accurate segmentation boundaries for foreground objects?
- How can we exploit the segment boundaries for computing temporally more consistent depth maps?

Once good quality depth maps for a given scene are obtained, those depth maps can be exploited to synthesize new views of the scene. This implies that the 3D scene is not altered, only the locations from which that scene is observed. More interesting editing operations would actually change the contents of the 3D scene. For the case of 2D images and video many such tools already exist. To extend 2D methods to 3D, one cannot simply apply the operation to the left and eye right images separately. Instead the computed depth should be taken into account. For example to ensure the correct relative size of objects at different depths, or the objects' orientation based on its location in the 3D scene. Another important aspect is for objects to maintain their stereo volume, to avoid them from appearing flat after editing.

The research questions we address related to 3D editing are:

- What level of quality should depth maps have?
- How can erroneous depth values in the depth map be handled in editing operations?
- How should depth be considered for editing 3D content?

The final stage of the pipeline after acquisition and editing, is the display of stereoscopic imagery. Different images for the left and right eye are displayed simultaneously, or near simultaneously, to invoke a depth perception. This approach is based on how the human visual system operates. However, there is an important difference. The eyes focus (accommodate) on a particular object in depth, and while simultaneously moving in opposite directions (vergence). When images for the left and right eye are displayed on the same plane in depth, i.e., the screen, this vergence–accommodation process has to be decoupled. This could result in visual fatigue [Hoffman et al., 2008].

It is therefore important to ensure the highest quality possible stereoscopic display. A particular problem we wish to address is that of crosstalk, also referred to as *ghosting*. Crosstalk occurs when the images for the left and right eye are not fully separated and as a result, the left eye can observe a dim copy of the image intended for the right eye, and vice versa. This is often the result of a physical property of the display system. Since these stereoscopic 3D display systems are observed by humans, compensation for crosstalk should exploit properties of the human visual system.

The research questions we address are:

- How can we formulate the problem and solve to obtain compensated imagery?
- What is a good model for crosstalk which enables to incorporate perceptual metrics?
- Which perceptual metrics should be considered for incorporation?

1.2 Contributions

In thesis we present the following contributions:

Multi-modal acquisition We describe an experimental acquisition system which combines a high quality central reference camera with satellite sensors. The sensors are (visible light) video cameras, Time-of-Flight depth camera, and a Long-wave (thermal) infrared camera. We also discuss the calibration for such a system.

Depth maps from multi-modal data We present a method to fuse the multi-modal data from the experimental system to compute depth maps.

We describe a local, per-pixel winner-take-all method that combines Time-of-Flight with photometric information from the video cameras. The satellite video cameras allow reasoning about occlusions in the scene, whereas the thermal camera provides segmentation information, especially for shots with human actors, which is often the case for live-action cinema and television. Together with a plane-fitting and trilateral smoothing step we achieve high quality results. Finally, we compare the contribution of each modality.

Segmentation and depth maps from multi-modal data Given the correlation between segmentation boundaries and depth contours, we describe a method which computes accurate segmentation boundaries for foreground objects in a video sequence. We describe how segmentation can be formulated as a labeling problem, which also takes temporal information into account. We demonstrate that by exploiting the multi-modal data and keeping a user in the loop, we can obtain accurate segment boundaries for challenging cases. A method for computing depth maps is presented which uses the segment boundaries, to obtain temporally more consistent depth maps.

Stereoscopic 3D editing —Stereo Copy & Paste We present a pipeline for stereoscopic 3D copy & paste editing. We describe the propagation of user performed segmentation on one image, to the other image for a stereoscopic pair. We use so-called stereo billboards, which approximate 3D objects with planar proxies and are robust to errors in depth maps. The planar proxies are computed using various constraints, including constraints based on stereoscopic parameters. Stereo billboards avoid the need for hole-filling when rendering composited results.

Ghosting compensation for stereoscopic 3D displays We present a general framework for compensating additive light pollution in display systems, which directly incorporates perceptual models. The compensation is formulated as an optimization problem, and we describe how the optimization can be made tractable. We describe how this framework is applied to compensate for ghosting, and also for scattering in concave displays. We achieve results which are preferred over results from other methods, which is verified by a user study.

1.3 Organization

The remainder of this thesis is organized as follows:

- In Chapter 2 we discuss the related work to the pipeline for stereoscopic 3D: acquisition, computing depth and segmentation, editing of stereoscopic 3D content, and stereoscopic 3D displays.
- In Chapter 3 we describe the experimental acquisition system we built which captures multi-modal data for aiding the computation of segmentation and depth maps.
- In Chapter 4 we discuss a method for fusing the data from the experimental system to compute depth maps, and compare the contribution of each modality from our experimental system.
- In Chapter 5 we discuss an interactive segmentation approach, and how the resulting segmentation is exploited in the computation of depth maps, to obtain temporally more consistent results.
- In Chapter 6 we discuss editing of stereoscopic 3D images for the application of copy and paste, and explain how the underlying depth of the scene is taken into account.
- In Chapter 7 we discuss a framework for compensation of ghosting in stereoscopic 3D display systems, and also discuss how the framework can be applied to more general light pollution problems for display systems.
- In Chapter 8 we provide a discussion on future work and conclusions on work presented in this thesis.
- In Appendix A we discuss a system using infrared structured light captured by multi-spectral cameras to compute depth maps, initially intended to obtain ground-truth data for comparisons.

1.4 Publications

This thesis is based in part on the following accepted peer-reviewed publications.

J. VAN BAAR, P. BEARDSLEY, M. POLLEFEYS, M. GROSS. Sensor Fusion for Depth Estimation, including TOF and Thermal Sensors *3DimPVT*, 2012.

J. VAN BAAR, P. BEARDSLEY, M. POLLEFEYS, M. GROSS. Interactive Video Segmentation Supported by Multiple Modalities, with an Application to Depth Maps *3DTV-CON*, 2012.

W.-Y. LO, J. VAN BAAR, C. KNAUS, M. ZWICKER, M. GROSS. Stereoscopic 3D copy & paste *Proceedings of ACM SIGGRAPH Asia 2010, ACM Transactions on Graphics, vol. 29, no. 6*

J. VAN BAAR, S. POULAKOS, W. JAROSZ, D. NOWROUZEZAHRAI, R. TAMSTORF, M. GROSS. Perceptually-based compensation of light pollution in display systems *APGV '11 Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*.

CHAPTER 2

Related Work

In this chapter we discuss the related work according to the different steps of the pipeline for stereoscopic 3D, as presented in Chapter 1. In Section 2.1 we discuss existing acquisition systems, and the system we propose instead. Section 2.2 gives an overview of various methods for computing depth maps, including methods which aim to fuse multiple modalities. Related work to video segmentation is discussed in Section 2.3. In Section 2.4 we discuss related work to editing of stereoscopic 3D content. Finally, related work to the compensation of ghosting in stereoscopic 3D displays, and compensation of global illumination in other types of displays is discussed in Section 2.5.

2.1 Acquisition

There are numerous existing systems for stereoscopic 3D acquisition, from research prototypes to commercial products. Research prototypes come in different varieties. Most prototype systems use a stereoscopic pair of cameras, or a trinocular camera configuration. Some systems integrate a camera, or stereoscopic pair of cameras, with additional modalities, such as Time-of-Flight depth [Zhu et al., 2008] or range sensing (LIDAR) [Diebel and Thrun, 2005]. Acquisition systems consisting of multiple (more than three) cameras

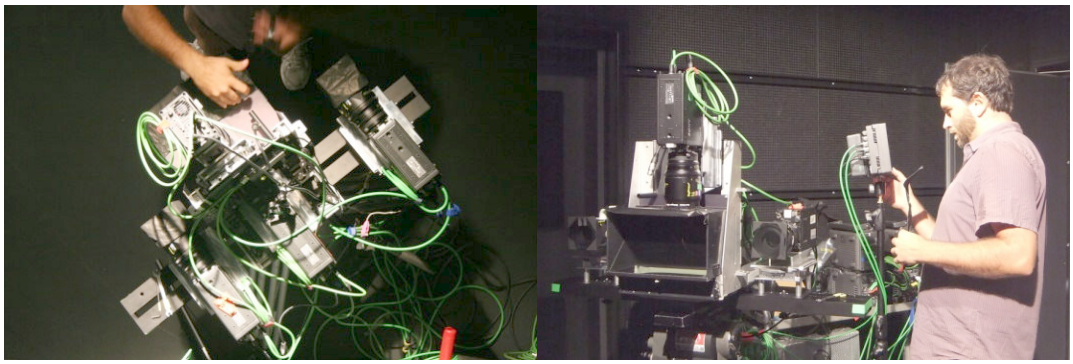


Figure 2.1: Multi-camera Rig. *Acquisition rig by HHI / fraunhofer. The rig consists of two cameras in a mirror configuration, with two satellite cameras on either side of the mirror.*

typically are static, and configured to capture some working volume, mostly for the purpose of full 3D reconstruction of objects [Seitz et al., 2006]. Some multi-camera systems on the other hand are configured for wide-baseline acquisition, for example a (static) system for view interpolation [Zitnick et al., 2004].

For commercial stereoscopic 3D rigs, the most common rigs use two cameras, either side-by-side or in a mirror configuration. The rigs require accurate motorized control of both their position and lens settings. More recently a multi-camera acquisition system was introduced which combines cameras in a mirror configuration with additional satellite cameras (see Figure 2.1). The motorized control makes the rigs rather bulky and wieldy.

Whether research prototype or commercial, the acquisition systems employ a homogeneous set of cameras, i.e. all cameras are of the same make and model. In some cases this leads to expensive and bulky systems. Furthermore, cinematographers and camera operators are trained to use a single camera for acquisition. We therefore propose instead to augment a single high quality reference camera with multiple satellite sensors. The sensors include (visible light) cameras, Time-of-Flight depth, and thermal (long wavelength) infrared. Our goal is to fuse the information from the various modalities for computing depth maps. We thus employ a heterogeneous set of cameras, where the satellite cameras are of lesser quality, and smaller form-factor. In Chapter 3 we describe the system in more detail, including the number of satellite cameras, expected benefits of each modality, and calibration for the system.

2.2 Processing—Depth Maps

Computing depth maps for two-view stereo or multi-view stereo has received, and continues to receive much research attention. It is beyond the scope of this thesis to provide a complete overview of the research in this area, but we discuss the most important related works. A taxonomy of different methods for depth (disparity) maps for two-view stereo is provided by Scharstein et al. [2002]. Ongoing evaluations of proposed approaches are published regularly [Scharstein and Szeliski, 2012]. A similar overview of multi-view stereo methods and evaluations are available [Seitz et al., 2006, 2012].

So-called global methods can obtain high quality results. Global methods express the problem of depth reconstruction as a graphical model, via a Markov Random Field, or the equivalent energy minimization formulation. A solution can then be obtained by performing inference on the graph. It has been shown that methods such as Graph Cuts [Boykov et al., 2001, Vogiatzis et al., 2007] and Belief Propagation [Sun et al., 2003, Felzenszwalb and Huttenlocher, 2006] can approximate global solutions. The drawback of global methods is their performance, since the problem size depends on the number of discrete labels and number pixels in the images, which is large for High-Definition resolution imagery. To avoid the long running times of global methods, Larsen et al. [2006] propose an iterative approach as an approximation to Belief Propagation. They back project the current depth hypothesis onto the cameras in a multi-view setup.

Sensor fusion has received a lot of attention in recent years, especially in combining visual data with depth measurements. Scanning approaches using laser range finders or Time-of-Flight depth sensors can reconstruct objects [Schuon et al., 2008] and even entire environments [Diebel and Thrun, 2005, Izadi et al., 2011]. The methods proposed by Schuon et al. [2008], Diebel and Thrun [2005], and Izadi et al. [2011] are based on a dense sampling of the object or scene, rather than using data at a single point in time. Long scanning times of the proposed methods do not permit dynamic objects or scenes. Some LIDAR systems allow real-time capture of dynamic indoor and outdoor environments [Velodyne, 2012]. Although these systems currently produce coarse 3D point clouds, the density is expected to increase for future releases of these scanning systems. For dense reconstruction of dynamic objects, Guan et al. [2008] use a combination of cameras and Time-of-Flight depth sensors. They formulate and solve a probability model for the occupancy within a bounded volume to obtain the 3D reconstruction. Their setup is thus limited to reconstruct objects within some bounded volume.

Tola et al. [2009] capture 3D video using a Time-of-Flight sensor to directly generate a 3D mesh based on the depth values from the depth sensor. Images from a registered camera are then used as textures for the 3D mesh. This approach does not fuse the information from the depth sensor and camera, and resulting depth maps are therefore noisy and low resolution. Some approaches attempt to increase the resolution from the Time-of-Flight depth sensors, either using implicit edge information by iterative bilateral filtering [Yang et al., 2007], or by more explicitly taking edge information into account [Lindner et al., 2008, Nair et al., 2012]. Time-of-Flight depth sensors have been used together with stereo for enhancing depth maps [Zhu et al., 2008]. The benefit of the fusion of depth measurements and stereo is shown for simple scenes with limited working volumes, rather than more natural and general scenes.

Many shots will contain one or more human actors, which in addition typically are foreground objects. Having accurate depth discontinuities for these foreground characters is important. Thermal infrared can provide segmentation information for human actors with respect to the background. Fusion with thermal infrared has been used in previous work. In [Conaire and Smeaton, 2008] the authors propose to exploit thermal infrared for tracking humans in video using so-called spatiograms. Their goal is the reliable tracking of occurrences rather than accurate segment boundaries. We instead exploit thermal infrared in the context of computing depth maps for general scenes.

Our goal is to compute high quality depth maps for general (indoor) scenes, typically involving multiple dynamic subjects. Depth maps are computed for a high resolution reference camera. We combine the information from multiple satellite cameras, with depth data from Time-of-Flight, and segmentation information from thermal infrared images. In Chapter 4 we propose a local method and show that we can obtain high quality depth maps from the fusion of these modalities.

2.3 Processing—Segmentation and Depth Maps

The computation of depth without consideration of temporal information could result in temporally inconsistent depth values. Temporal information can be incorporated by extending graphical models to contain nodes from previous and next frames using optical flow information [Larsen et al., 2007]. Temporally more smooth depth values may be obtained by reprojection of depth values to adjacent frames followed by bundle adjustment [Zhang et al.,

2009]. Yang et al. [2012a] show that background and motion models can be recovered in multiple steps, and subsequently used to compute temporally more smooth depth values.

Depth and color discontinuities typically occur at object boundaries, and are therefore correlated. Information about the object boundaries could therefore help to improve depth maps. Object boundaries information over time could help to improve the temporal consistency of depth maps. We thus aim to accurately segment foreground objects from the background for the frames in a video sequence.

Video segmentation is a well-studied research area. Previous work can be classified as either interactive or entirely automatic. Interactive segmentation methods require the user to provide initial scribbles to indicate fore- and background. Local color models are learned from the initial indication, and pixels are assigned a label according to these models. The method proposed by Rother et al. [2004] is intended for still images and uses iterative graph cuts for obtaining an accurate segmentation and matting. Some methods for video segmentation [Bai et al., 2009, Gong and Cheng, 2011, Price et al., 2009] incorporate optical flow to propagate the segmentation to subsequent frames. Price et al. additionally incorporate corrections made by the user. Temporal information over the video sequence can be encapsulated into point-trajectories for a sparse set of image samples [Ochs and Brox, 2011]. The sparse samples are then interpolated to obtain a per-pixel segmentation. Although the methods mentioned above can obtain accurate segment boundaries, they are limited to segmentation of *single* foreground objects only.

The goal of most automatic video segmentation methods [DeMenthon and Megret, 2002, Paris, 2008, Grundmann et al., 2010, Vazquez-Reina et al., 2010, Lezama et al., 2011] is to obtain temporally consistent segmentations. Segmentation boundaries are typically inaccurate. Since these methods are targeted for applications such as object tracking, or non-photorealistic rendering, accurate segment boundaries are not crucial. These methods support the segmentation of multiple objects. Objects that occlude each other in the video sequence, cannot be consistently segmented. The methods proposed by [Grundmann et al., 2010, Vazquez-Reina et al., 2010] consider video segmentation to be a labeling problem, and formulate this as an energy minimization problem. Both methods rely on a hierarchical segmentation, where images are progressively segmented from coarse to fine segments. Grundmann et al. rely on optical flow between subsequent frames for grouping corresponding segments temporally. On the other hand, Vazquez-Reina et al. avoid the need for optical flow and consider (coarse) overlapping segments to correspond temporally. In addition, for this approach information

across multiple frames of the video sequence is incorporated as higher-order clique terms in a graphical model.

The correlation between depth and color discontinuities is used for spatially-adaptive weighting of the smoothness term in global optimization methods, which aims to avoid smoothing over depth boundaries [Felzenszwalb and Huttenlocher, 2006], or object boundaries [Boykov and Funka-Lea, 2006]. In [Zitnick and Kang, 2007b] the authors propose to oversegment images of video sequences and compute depth maps with consistent object boundaries. The correlation is also used to simultaneously compute segmentation and depth maps [Bleyer et al., 2011, Zhang et al., 2011]. These methods operate on stereo image pairs, and the resulting segmentation is used to handle stereo occlusions, as opposed to object occlusions. Background segmentation is exploited in [Yang et al., 2012a] for improving depth maps.

We propose an interactive approach to accurately segment multiple (possibly occluding) objects in a video sequence. These accurate boundaries are then exploited as explicit constraints when computing depth maps. We define the video segmentation problem as an energy minimization problem to propagate known segmentations to the intermediate frames of a video sequence. Our approach is interactive: the user provides an initial segmentation for the first and last frame of a video sequence, and supervises the propagation to ensure correct labeling. The pixels in each video frame are clustered into regions called superpixels, and the task therefore becomes to label the superpixels according to the segment they belong to. To obtain accurate segment boundaries, we employ a boundary refinement step based on local color models near the initial boundaries. We show that the Time-of-Flight depth can be exploited to reduce the size of the problem. In addition, previous approaches rely only on color information and fail in areas with lack of (color) contrast, such as textureless areas, or in areas of similarly colored fore- and background objects. We show that thermal infrared can improve the result in such situations.

2.4 Stereoscopic Editing—Stereo Copy & Paste

Computing depth maps and performing segmentation are necessary operations for editing stereoscopic 3D contents. Our focus will be on copy & paste for stereoscopic images. The methods we present in this thesis for computing depth, are specifically designed to operate on multi-modal data. However, for stereoscopic copy & paste, we would like to be able to process stereoscopic image pairs in general. We thus choose an existing method for com-

puting depth [Smith et al., 2009] instead. Since the depth maps likely contain inaccuracies, we aim for a system that is robust with respect to inaccuracies in the depth map.

To support stereoscopic copy & paste, we have to accurately segment multiple objects, either in still images, or in a video sequence. The interactive segmentation approach we present in this thesis related to computing depth maps, could be used in this context. In any case, this interactive segmentation approach requires an initial segmentation for the first and last frame of a video sequence. We would like to obtain these segmentations with the least amount of user input. Methods which are designed for segmentation of single objects, such as the iterative Graph Cuts approach [Rother et al., 2004], or the incremental Graph Cuts scheme [Liu et al., 2009b], would require a significant amount of user interaction to segment multiple objects. Lu et al. [2007] describe a multi-class segmentation method, but this can handle only a small number of distinct classes and is computationally expensive. To allow for easy multiple object segmentation we combine the fast cluster-merging method by Ning et al. [2010], with mean-shift clustering [Comaniciu and Meer, 2002].

We would like to segment objects for a stereoscopic pair of images. To avoid having the user perform the segmentation twice, we aim to propagate the segmentation for one eye image to the other eye. This propagation is related to cosegmentation. Cosegmentation aims at segmenting the common parts between a pair or a sequence of images. Rother et al. [2006] exploit histograms for consistency between foreground objects in images. Dong Seon and Figueiredo [2007] encode the consistency between objects in frames within a prior and solve a mixture model. Zitnick et al. [2005] aim for consistent segmentation and motion simultaneously, using segment shape and optical flow between images as constraints and finally solving an energy minimization problem. Motion, optical flow, and tracking have also been proposed in segmentation propagation for video sequences [Chuang et al., 2002, Agarwala et al., 2004]. Rather than relying on multiple frames, or modeling the consistency between objects explicitly, we have chosen to adopt Video Snapcut [Bai et al., 2009] which propagates a set of local windows along the segmentation contour with associated color information.

Copy & paste editing for 2D images has received much attention recently. Poisson image editing [Pérez et al., 2003] has the advantage that no accurate segmentation is necessary, but requires care to be taken to avoid smearing in the case of dissimilar backgrounds. Drag and Drop Pasting [Jia et al., 2006] attempts to avoid smear by computing an optimal boundary for Poisson blending. However this method still will not produce desired results for multiple

(partially occluding) objects of different textures. Alpha matting [Wang and Cohen, 2008] on the other hand will be able to handle such cases, and we compute alpha mattes for all segmentations in our system.

In Photo Clip-Art [Lalonde et al., 2007] objects are inserted into a target image from a database of pre-segmented and labeled images. The 3D scene structure, and lighting are estimated by image analysis and to determine which object to retrieve from the database. In our case the user explicitly selects the objects to be copied and pasted for stereoscopic 3D images, and we address the challenges that arise with this.

Several stereoscopic editing approaches exist. Stereoscopic Inpainting [Wang et al., 2008] describes a segmentation-based method which exploits disparity maps to fill in missing depth and color due to occlusion in stereoscopic images. Editing methods for manipulating stereo parameters, e.g., stereo baseline, compute disparity maps to adjust the parameters locally or globally [Lang et al., 2010, Koppal et al., 2010, Wang and Sawchuk, 2008]. A commercial stereo editing tool we are currently aware of is the Ocula plug-in for Nuke [The Foundry, 2012]. The focus in these methods is either on foreground object removal, color correction, alignment correction, or stereo view synthesis rather than object copy & paste.

Rhee et al. [2007] introduced the concept of stereo billboards as planar proxies for stereoscopic telepresence display under the assumption that objects are always humans and fronto-parallel to the camera. In contrast, our stereo billboards represent arbitrary 3D objects, and the optimal orientations are computed using the objects' reconstructed 3D points as constraints.

The system we will present shares some similarities with pop-up light fields [Shum et al., 2004], which is an image-based rendering system that models a sparse light field using a layered representation. In this system, the user interactively segments layers for *pop-up*, with the goal of high quality rendering of a sparse light field. In contrast, our system allows to copy objects from different sources, and is not a layered representation.

2.5 Stereoscopic Display—Deghosting

The illusion of stereoscopic depth is dependent on a viewer's ability to fuse corresponding features or edges presented to the two eyes [Howard and Rogers, 2002]. Viewing stereoscopic images is a demanding task for the HVS. It requires a decoupling between focus and eye vergence that has been

demonstrated to influence not only viewer discomfort, but also hinder visual attention and depth discrimination [Hoffman et al., 2008].

The perception of ghosting can be considered a “binocular noise" that further hinders fusion limits and visual comfort. Yeh and Silverstein demonstrated that crosstalk significantly influences the ability to fuse widely separated images via binocular eye vergence movement [1990]. Ghost images may introduce unintended edges and binocular rivalry making visual processing unstable, unpredictable, and impair guiding visual attention [Patterson, 2007]. It has also been found to inhibit the interpretation of depth [Tsirlin et al., 2011].

Use of even minimal crosstalk has been found to strongly affect subjective ratings of display image quality and visual comfort [Yeh and Silverstein, 1990, Kooi and Toet, 2004]. Although acceptable crosstalk may generally be as high as 5-10%, the detection and acceptability thresholds can be significantly reduced with higher image contrast or larger disparity [Wang et al., 2011]. There is a strong need to remove the detection of crosstalk.

Subtractive compensation methods for active (time-sequential) and passive (light modulation) stereo display systems, subtract the predicted ghosting contribution prior to display [Konrad et al., 2000, Klimenko et al., 2003]. These methods assume that there is sufficient *signal* to subtract from since physical systems cannot inject negative light. To fully compensate in these cases, the black level is raised globally (automatically), or locally (manually). Smit et al. [2007] proposed a perceptually motivated extension to subtractive compensation. They perform subtraction in the perceptually uniform *CIE-Lab*, instead of *RGB* color space. This results in less visible ghosting compared to standard subtractive methods. However, in the case of low luminance, their method suffers from the same problem as other subtractive methods, and leaves ghosting as uncorrectable. We specifically want to address these “uncorrectable” cases, and instead propose a perceptually-based distribution of the ghosting error to reduced sensitivity regions of the HVS.

Perceptually-based methods have been extensively used and an exhaustive list would be beyond the scope of this thesis. We discuss the most relevant works. Perceptual models have been exploited to determine if the texture of objects masks an underlying coarse tessellation [Ferwerda et al., 1997], and in stopping criteria for global illumination [Ramasubramanian et al., 1999, Mantiuk et al., 2006, Longhurst et al., 2006, Sundstedt et al., 2007]. We explicitly exploit perceptual models for formulating an optimization framework.

Tone mapping involves the display of an image with a higher dynamic range on a display with a lower dynamic range [Reinhard et al., 2002, Mantiuk

et al., 2008]. The method that we propose is related to local tone mapping in that our goal is to take an image with a higher dynamic range (the intended image), and display it on a (locally) lower dynamic range display (due to the light pollution). We aim to distribute the error smoothly in a local region, and we propose to exploit HVS properties to do this in a perceptually more optimal manner.

Majumder and Stevens [2005] aim to obtain a global smoothly varying luminance in a multi-projector display by incorporating perceptual metrics. Grosse et al. [2010] exploit the CSF to precompute a binary mask, which in turn is used to compute an optimal coded aperture. Our work differs in that our goal is to locally compensate for ghosting only. This requires a different formulation for the optimization problem which we describe in Chapter 7.

In image processing domain, perceptual metrics are incorporated into a Visible Difference Predictors (VDP), which aims to quantify the perceptual difference between a reference and a test image [Daly, 1992, Lubin, 1995]. Furthermore, Nadenau et al. [2001] propose to exploit the contrast sensitivity function (CSF) for weighting the coefficients of a wavelet decomposition at different levels. We propose to incorporate the CSF and components of the VDP directly in our optimization framework.

Several subtractive compensation methods have been suggested to compensate for indirect scattering by modifying the image before projection [Bimber et al., 2006, Mukaigawa et al., 2006, Wetzstein and Bimber, 2007, Dehos et al., 2008]. Bimber et al. [2007] provides a more comprehensive overview of other related work in this field. All these methods suffer from the same problems as subtracive deghosting methods. We propose an analytic subtractive compensation for spherical domes, that operates on full-resolution images, and combine this with our perceptual framework to redistribute this error into visually less important regions.

CHAPTER 3

Multi-Modal System for Acquisition

In this chapter we describe the experimental acquisition system we built. We describe both the sensors that are used in the system, and the calibration procedures we employed. The high level description is a system consisting of a central high quality reference camera augmented with satellite sensors. The system combines sensors of various sensing modalities: color (visible light), Time-of-Flight depth and long-wave infrared. We discuss the two types of calibration we employ: geometric and photometric. Geometric calibration determines the lens parameters, location and orientation of the satellite sensors with respect to the reference camera. The photometric calibration aims to find a color transform for each camera to ensure a similar photometric response among cameras. In addition to the experimental acquisition system we also describe a system initially aimed to capture ground truth data. This system can simultaneously capture a scene that is captured with our experimental acquisition system.



Figure 3.1: Compact Cinema Cameras. *Some examples of compact, full frame imaging sensor, cinema cameras:* ***Left*** *the ARRI Alexa M,* ***Middle*** *Silicon Imaging 2K mini,* ***Right*** *Blackmagic Design Cinema Camera.*

3.1 Introduction and Motivation

To generate high quality content, for example for cinematic motion pictures, a high quality camera is required. High quality cameras mostly distinguish themselves from low quality ones in terms of sensitivity and noise. Cameras with higher sensitivity are able to capture a higher dynamic range of intensities. Sensitivity is important to ensure a high signal-to-noise ratio even for low luminance areas in the scene. High quality cameras consist of high-end components and are therefore expensive. In addition, the physical footprint of high quality cameras tends to be large compared to that of lower quality ones, although we are beginning to see more compact high-end cameras (see Figure 3.1 for examples).

For capturing stereoscopic content, typically two cameras are used. Having additional cameras beyond two, helps to improve the quality of computed depth maps. Replicating the high quality reference camera to create a homogenous multi-camera system could result in a bulky, expensive system. More importantly however, cinematographers and camera operators are used to capture a scene with a single camera. It would be desirable for the cinematographer and camera operators to not be encumbered by additional capturing devices, and instead focus only on acquisition with the single camera.

As already discussed in Chapters 1 and 2, computing depth maps based on color information alone could lead to ambiguities. We therefore would like to capture additional information besides color images. We propose to incorporate two additional modalities: Time-of-Flight depth and thermal (long-wavelength) infrared. Since there are no readily available experimental systems, we built our own.

We propose to augment a central high quality camera with several satellite sensors. The satellite cameras will be of less quality compared to the central camera. We aim to exploit current hardware advances: lower cost, smaller form factor but increasing quality. For example, small USB board cameras have a weight of 30g and dimensions of a couple of cm. New low-end thermal cameras are a similar size and price to machine vision cameras [Infrared Cameras Inc., 2012]. Building a multiple sensor modalities single system is thus becoming practical. Our experimental setup is merely a prototype. It is feasible to eventually envisage a compact cinematographic camera augmented with satellite sensors with little impact on its normal handling and workflow.

3.2 System

Figure 3.2 shows a frontal view of the experimental rig we built. In the center (mostly obscured behind the beamsplitting glass) is the reference camera (A). On each side of the reference camera are two satellite cameras (B). The Time-of-Flight camera (C) is mounted just above the reference camera. Finally, the thermal infrared camera (D) is in a beamsplitting configuration with the reference camera.

The beamsplitter consists of thermally coated K-glass at a 45° angle. All of the visible light is allowed to pass through to the reference camera, whereas most of the thermal radiation is reflected towards the thermal camera. This configuration allows the reference camera and the thermal camera to have nearly the same optical path, and accurate alignment. Next, we briefly describe each hardware component in more detail, the positioning of the sensors, and synchronization.

3.2.1 High Quality Reference Camera

The definition of high quality for cinematographic applications may be different from the definition of high quality for broadcasting. We therefore do not quantify what high quality is, but rather use the term more loosely. For our experimental system we chose a Sony PMW-350K camera. This camera has three different CMOS sensors for red, green and blue. Before the incoming light is recorded by the sensors, the light is first split by a special prism into three bands of wavelengths representing red, green and blue light. The camera progressively captures images of 1280×720 resolution. From here on we refer to this camera as the reference camera.

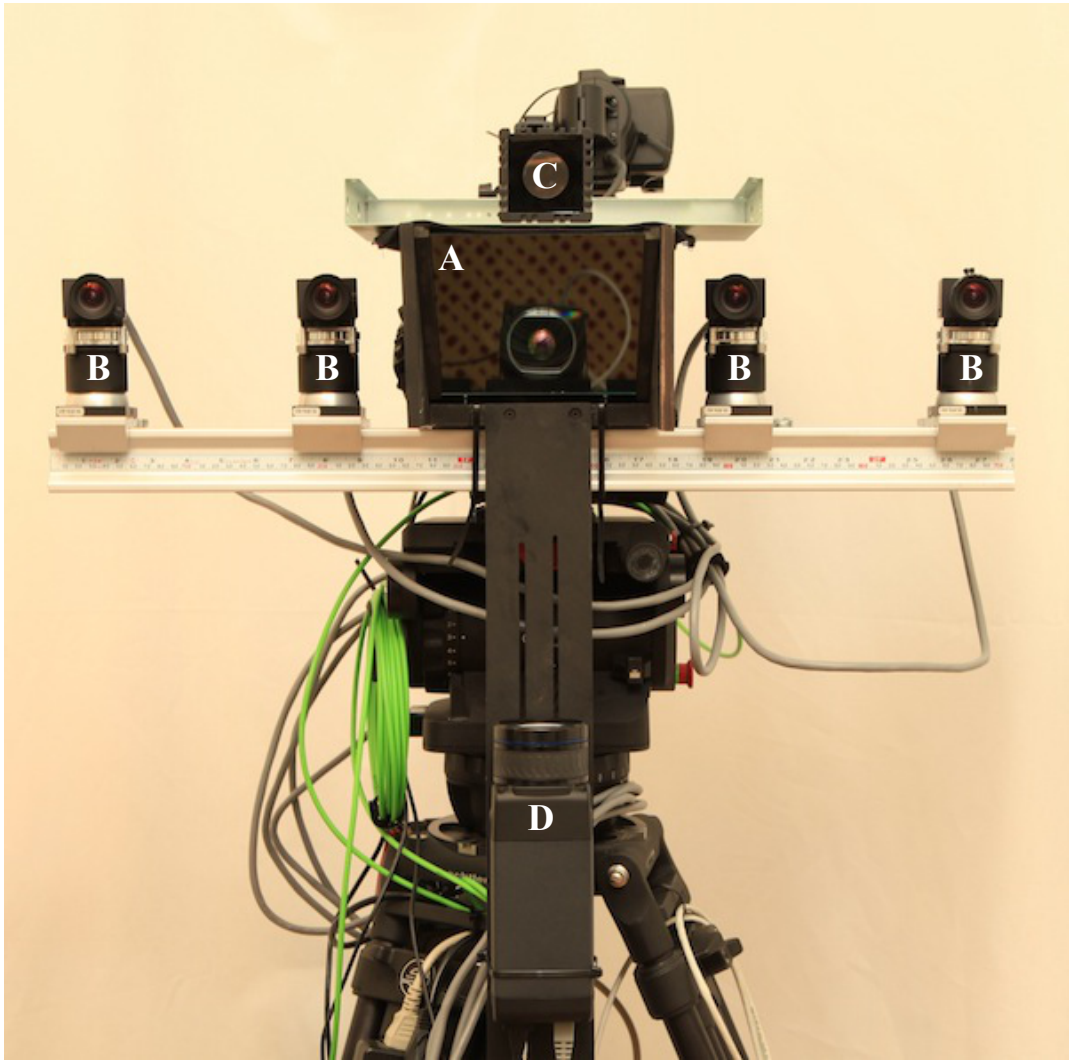


Figure 3.2: Experimental Acquisition System. *A The high quality reference camera.* ***B*** *The satellite cameras.* ***C*** *The Time-of-Flight depth camera.* ***D*** *The thermal infrared camera.*

3.2.2 Satellite Cameras

For the satellite cameras we chose PtGrey Grasshopper cameras. The cameras capture images of 1600×1200 resolution. Although the resolution is larger compared to the reference camera, the imaging sensor is considerably smaller. The viewing angle for the satellite cameras compared to the reference camera is larger. This ensures that the area of the scene imaged by the reference camera fits within the area imaged by the satellite cameras. The quality of the lenses we use for the satellite cameras compared to the lens for

the reference camera is also lower, based on a comparison of the corresponding Modulation Transfer Function (MTF) charts.

The satellite and reference cameras form a heterogeneous set. As we will discuss in Chapter 4, we will have to take this fact into account when determining color consistencies between a satellite camera and the reference camera.

3.2.3 Time-of-Flight Depth Camera

The Time-of-Flight depth camera we use in our experimental system is the Mesa Imaging SR 4000. Our choice of depth camera is mostly practical, since the SR 4000 can be triggered via external trigger input. However, other depth cameras are available as well, most notably the Microsoft Kinect. We discuss this in more detail in Chapter 4.

The SR 4000 camera measures depth for 176×144 image pixels. The camera has 24 infrared LEDs positioned in a circle around the imaging sensor. The basic principle of Time-of-Flight is to measure the time it takes for a pulse of light to be reflected and recorded by the imaging sensor. Depending on the modulation frequency of the active illumination, the range of the camera can be either 5m or 10m. Although the absolute (internal) accuracy is specified as 10mm, the actual accuracy of the depth measurements depends on an objects' distance from the camera and its IR reflectivity.

Time-of-Flight depth is less reliable near depth contours where the IR light strikes an oblique surface, and most of the light is not directly reflected back to the camera. As a result, the depth contours in Time-of-Flight cameras typically do not correspond with the actual physical discontinuities. In general the reflectivity properties of materials determine the amount of noise present in the measurements. In addition, a depth value is typically computed over several samples of the signal which represents the reflected light. Moving objects therefore introduce additional noise for depth values close to their depth contours.

3.2.4 Far Infrared Thermal Camera

The thermal infrared camera is a Flir SC645 camera. This camera records images at 640×480 resolution. The image sensor can passively measure thermal radiation using an uncooled microbolometer operating in the wavelength range of 7.5–14 μm. The SC645 can be thermally calibrated to an accuracy of

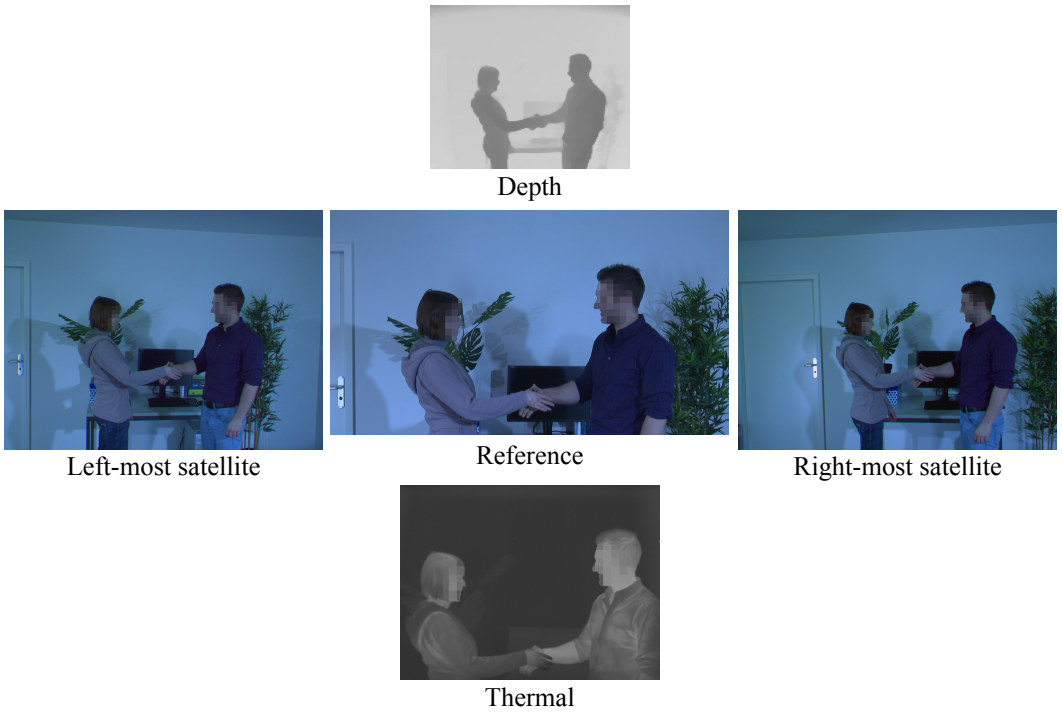


Figure 3.3: Example Dataset. *Example images from different cameras at one time instant.* **Top -** *depth camera.* **Center -** *cinematographic camera.* **Left and right -** *the outermost satellite cameras.* **Bottom -** *thermal camera.*

about 2°. However, we are only interested in relative measurements. The relative pixel-to-pixel accuracy is less than 100 mK for room temperature measurements.

Figure 3.3 shows an example of the images acquired with the different sensors of our experimental system. For simplicity, we only show example images for the left- and right-most satellite cameras (labeled (B) in Figure 3.2).

3.2.5 Sensors Positioning

The satellite cameras are positioned to each side of the reference camera to create a wide-baseline setup. The size of the baseline could vary depending on the scene that is being acquired. A larger baseline for scenes with objects that are farther away from the camera, a smaller baseline for scenes with objects closer to the camera.

3.2.6 Synchronization

In the most ideal situation, the images for the different sensors are captured at a single instance in time. This requires synchronization of the different sensors with an external trigger signal. The satellite cameras and Time-of-Flight camera have built-in capability to be synchronized via an external trigger signal. However, synchronization of the reference camera and thermal camera proved difficult. The reference camera requires a so-called Genlock reference video signal for synchronization, which should then also be synchronized with the trigger signals. The thermal camera does not support synchronization at the level of image capture. Instead, a trigger signal can be used to start or stop continuous image capture.

In order to capture images with all sensors as close to a single instance in time as possible, we implemented the following scheme. All sensors in the system are able to capture images at a framerate of 25 frames per second (fps). We use an off-the-shelf FPGA which can be programmed to generate the desired trigger signal [CESYS, 2012]. The trigger signals are then constructed such that they represent acquisition at a framerate of 25 fps. The reference and thermal camera are also set to acquire images at a framerate of 25 fps.

Although we minimize the difference in time at which the images are captured for all sensors, without explicit synchronization fast motions could lead to an object being captured at different locations in space. This in turn will impact the quality of the results. We therefore aim to avoid fast motion when capturing data with our experimental system.

3.3 Multi-Sensor Multi-Modal Acquisition

3.3.1 Expected Benefit of Satellite Cameras / Sensors

The satellite cameras can be exploited to reason about occlusions in the scene. Regions in the scene are occluded depending on the viewpoint from which the image was captured. We can compare the contributions from multiple satellite cameras on one side of the reference camera. If the contributions are very different we may be dealing with an occlusion region.

3.3.2 Expected Benefit of Adding ToF Depth Sensor

The Time-of-Flight depth sensor directly records depth values. The resolution is small and some details are not captured in the recorded depth image. However, in other cases the depth sensor records correct depth values. This will help obtain correct depth values in uniform color regions, and repeated texture regions. It could also help in regions of occlusions.

3.3.3 Expected Benefit of Adding Thermal Sensor

Many video sequences have live actors. Thermal sensors are particularly useful in this case, as the thermal signature of live actors will typically be different from inanimate objects and backgrounds. Even in some cases the thermal signatures of different actors may be different. This information can be exploited as a first order prior on the segmentation, much like the red, green and blue channels are used.

3.4 Calibration

3.4.1 Geometric Calibration

In order to determine corresponding pixels between cameras, the cameras should be accurately geometrically calibrated. Geometric calibration determines the camera pose (extrinsics), and lens parameters (intrinsics) for each camera.

The extrinsics consist of a rotation and a translation. Given a 3D point $\mathbf{X} = [XYZ]^T$ in some coordinate space. The 3D point $\mathbf{X}$ relates to a pixel $\mathbf{u} = [u, v]^T$ in the camera by the following equation:

$$\mathbf{x} \cong \mathbf{PX} = \mathbf{K}[\mathbf{R}|\mathbf{T}]\mathbf{X}, u = x/z, v = y/z. \tag{3.1}$$

The matrix $\mathbf{K}$ is a 3×3 intrinsics matrix containing focal length, and principal point. In our case we compute the focal length separately for the x- and y-directions. The principal point is a 2D point on the image plane, representing the projection of the optical axis onto the image plane [Hartley and Zisserman, 2004]. Real lenses usually introduce distortion. We use a five parameter

lens distortion model [Zhang, 2000], which accounts for radial and tangential distortion:

$$u' = u(1 + k_1 r + k_2 r^2 + k_3 r^4) + [2p_1 uv + p_2(r + 2u^2)], \quad (3.2)$$
$$v' = v(1 + k_1 r + k_2 r^2 + k_3 r^4) + [p_1(r + 2v^2) + 2p_2 uv]. \quad (3.3)$$

Here r is determined as $u^2 + v^2$. The k_3 parameter is usually set to zero. Our geometric calibration procedure then consists of the following steps:

1. Present a series of checker patterns at various poses. The corners of the checker pattern are detected for each pose.
2. Compute lens parameters for the satellite and reference cameras.
3. Compute the extrinsics between:
   a) Each satellite camera and the reference camera.
   b) The Time-of-Flight camera and the reference camera.
4. Compute homography between the thermal camera and the reference camera.

Satellite Cameras The calibration of lens parameters and initial pose between each satellite camera and the reference camera is performed using known techniques [Bouguet, 2012]. We then perform bundle adjustment [Lourakis and Argyros, 2009] as a final step.

Time-of-Flight Camera The focal length, principal point and lens distortion parameters for the Time-of-Flight camera are calibrated during manufacturing and they can be retrieved from the camera. These parameter values are used by the corresponding software library for the Time-of-Flight camera to compute per-pixel X, Y, Z coordinates.

The Time-of-Flight depth camera produces X, Y, Z coordinates measured in meters. The Time-of-Flight depth camera and the reference camera are thus related by a transformation and scaling. Given the detected feature point in the Time-of-Flight camera image, $\mathbf{u}_{ToF}$, we can lookup the corresponding 3D measurement $\mathbf{X}_{ToF} = (X, Y, Z)^T$. This gives a set of corresponding 3D points between the reference camera, $\mathbf{X}_{ref}$ and the Time-of-Flight camera, $\mathbf{X}_{ToF}$. The goal is now to compute the transformation $Tr_{ToF \to ref}(\mathbf{X}_{ToF}) = \mathbf{X}_{ref}$.

First, we compute the closed-form solution for the rigid motion between $\mathbf{X}_{ref}$ and $\mathbf{X}_{ToF}$ [Horn, 1987]. Rigid motion consists of a uniform scale s, rotation R,

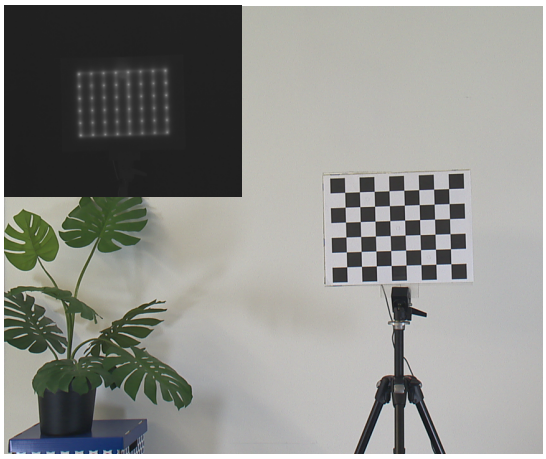


Figure 3.4: Thermal Camera Calibration. *Resistors are accurately mounted at the corners of the checker pattern. By applying a current to the resistors, they will heat up. Shown are the image acquired by the reference camera, and (**inset**) the image acquired by the thermal camera. During calibration the centers of each camera are first aligned manually, and then a homography is computed to warp the thermal image for accurate final alignment.*

and translation T. We use the closed-form solution as the starting point for a non-linear optimization. Rather than a uniform scaling, we estimate different scaling parameters for X, Y, Z. $Tr_{ToF \to ref}$ therefore consists of 9 parameters:

$$\mathbf{X}_{ref} = Tr_{ToF \to ref}(\mathbf{X}_{ToF}) = diag(s_X, s_Y, s_Z)\mathbf{R}\mathbf{X}_{ToF} + \mathbf{T}. \tag{3.4}$$

The non-linear optimization then aims to simultaneously minimize the distance between the 3D points, the reprojection error for the reference camera image, and the reprojection error for the ToF camera.

Thermal Camera The goal for the thermal camera is to align it's center of projection to the center of projection of the reference camera. An initial alignment between thermal and reference camera is achieved by manual adjustment. Final alignment is then achieved by warping the thermal camera image using a homography computed between the thermal and reference camera. The homography is computed based on a set of corresponding 2D features between the thermal and reference images. In order to simultaneously detect visible features in the reference image, and thermal features in the thermal image, we precisely mounted resistors at each checker corner location (see Figure 3.4). By applying a current on the resistors they will heat up, and we



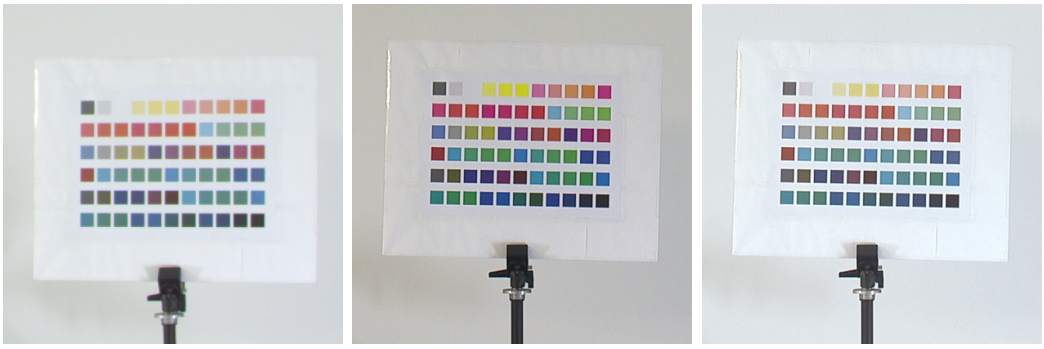
Figure 3.5: Thermal Camera Calibration Result. ***Left*** *Thermal camera image.* ***Middle*** *Accurately aligned thermal image superimposed on the reference camera color image.* ***Right*** *Reference camera image. The thermal signal of humans is high compared to background objects, even when covered by clothing.*

compute the first-order moment for each blob to determine its center with sub-pixel precision. The images retrieved from the thermal camera are already corrected for distortion, and for computing a homography we do not need any additional lens parameters.

An example of the registration between the reference RGB image and the thermal image is shown in Figure 3.5, with the thermal image superimposed on the color image.

3.4.2 Photometric Calibration

To compute photometric calibration we acquire an image of a color pattern with all cameras. We do not aim to compute absolute photometric calibration, but instead relative photometric calibration to one of the cameras in the experimental system. A user has to click the centers of the four outermost color patches for each camera. The centers for the remaining color patches are then determined automatically. For each camera we compute the average photometric response over a small region around each color patch center. Corresponding responses between two cameras are then used to compute a polynomial of degree two for each color channel separately [Ilie and Welch, 2005]. Figure 3.6 shows the color pattern acquired by two cameras, and the result of the computed photometric transform.



(a) (b) (c)

Figure 3.6: Photometric Calibration. *The color pattern is acquired by all cameras in the experimental system. We transform the color spaces to obtain a similar photometric response for each camera.* ***(a)*** *Image used as the photometric reference image.* ***(b)*** *Image acquired by the reference camera prior to transform.* ***(c)*** *Reference image after photometric transform.*

3.5 Discussion

In Chapters 4 and 5 we will describe how we can exploit multiple modalities to compute depth maps and perform interactive segmentation, using the multi-modal acquisition system we described in this chapter.

CHAPTER 4

Processing—Depth Maps

Computing depth maps for acquired videos is an important operation which serves as a first step for many subsequent processing steps, for example the insertion of Computer Graphics elements into a video. Our focus application is the computation of depth maps for live-action 3D cinema and television, i.e. television programs which are not broadcast in real-time or near real-time, such as sport games. High quality depth maps are required for being able to edit the stereoscopic images that will be displayed, based on the 3D composition of the underlying scene. We are therefore mostly concerned with the correctness of depth maps, and less concerned with performance aspects.

In this chapter we describe a method to compute depth maps using our experimental system presented in Chapter 3. The experimental system consists of a high quality, central reference camera augmented with satellite sensors of different modalities (see Figure 3.2). The goal with this system is for cinematographers to capture scenes the way they are used to, using a single camera, and not be encumbered by the satellite sensors. The additional satellite sensors merely capture data to aid the computation of depth maps for the acquired scenes.

We will present results we obtain with our method, and compare them to results that can be obtained with other approaches. The method presented in this chapter operates on individual frames in video sequences. As a con-

sequence, the depth maps between subsequent frames vary slightly. In the next chapter we present a method which aims to compute temporally more consistent depth maps.

4.1 Motivation

The goal is to obtain high-quality depth maps. Computing accurate depth maps for general scenes remains a difficult problem. Inferring depth based on photometric information alone could result in ambiguities. Existing work has shown the advantage of active sensors for measuring 3D information, including Time-of-Flight sensors and structured illumination. The Kinect [Microsoft, 2012] has been a landmark device in bringing 3D sensing to the mass market, using an infrared illumination source that projects a structured speckle pattern onto the scene. But there are limitations with both approaches —Time-of-Flight response falls off on oblique surfaces, as found for example near the occluding contour of a curved object. While structured illumination, such as the speckle pattern for Kinect, delivers sparse information, not pixel-dense measurements. For both approaches, when the surface is poorly reflective with respect to the infrared illumination, the depth values will become unreliable.

Since our focus is on live-action cinema and television, we expect many of the shots to contain human actors. Thermal cameras detect emitted heat and human skin, even when covered by some layers of clothing, which typically gives a thermal gradient with the background. A beam splitter is used to capture registered images for the reference camera and the thermal camera. Beam splitting with two visible light cameras has a disadvantage that only half the light enters each camera. But beam splitting for a visible light camera and a thermal camera using a K-glass beam splitter results in most of the incident visible light entering the visible light camera, and most of the thermal radiation entering the thermal camera.

We compute depth maps by combining the information from the various sensors, a process we refer to as sensor fusion. The first contribution of this chapter is to demonstrate how Time-of-Flight data is combined with data from multiple cameras to compute high quality depth maps. The depth map is actually computed for the (high-resolution) reference camera, supported by the lower resolution Time-of-Flight sensor and support cameras. The method utilizes support cameras on both sides of the reference camera in order to do explicit reasoning about occlusions. We demonstrate that by using a local method in combination with plane-fitting we can obtain high quality results

for challenging scenes. In addition, we demonstrate that the occlusion reasoning in combination with a global method also produces high quality results, albeit for longer processing times. Our definition of local and global is provided in Section 4.2. We also discuss how we exploit the fact that thermal sensing aids segmentation by detecting thermal gradients between humans and the background.

Our second contribution is to demonstrate the advantage of sensor fusion including thermal sensing. We analyze several examples of cases that might typically occur in scenes with human actors. We compute depth maps using only photometric information, and compare them to depth maps which are computed including the contribution from the Time-of-Flight depth, the thermal signal, and their combination.

4.2 Problem Formulation

Depth can be computed for a minimum of two images of the same scene acquired from different viewing locations. The cameras which acquire the images are assumed to be calibrated for both intrinsics and extrinsics. The images are then rectified such that the epipolar lines are horizontal for both images, and the epipolar line in one image directly corresponds with the same epipolar line in the other image [Hartley and Zisserman, 2004]. One of the images is then chosen as the reference image. For a given pixel p along the epipolar line in the reference image, the corresponding pixel p' along the epipolar line in the other image is chosen as the one with maximum color similarity. The horizontal difference between p and p' is denoted as disparity. The process is repeated for all pixels in the reference image and results in a so-called disparity map. Given that disparity is inversely related to depth, i.e., pixels at greater depth have smaller disparities, disparity maps may be referred to as depth maps as well.

The minimum and maximum disparities are dependent on the depth layout of the underlying scene. Therefore, when searching for a corresponding pixel along an epipolar line, only *discrete* disparity values in the range between the minimum and maximum disparity are considered. The collection of similarity values for each disparity within the disparity range, and for all pixels in an image is called a disparity space image (DSI) [Scharstein and Szeliski, 2002].

Each discrete disparity value in the disparity range can be considered a label. Computing the disparity map can thus be considered a labeling problem of

the set of image pixels $\mathcal{P}$ given a set of labels $\mathcal{L}$. A labeling f then assigns a label $f_p \in \mathcal{L}$ to each pixel $p \in \mathcal{P}$.

The quality of a labeling can be given by an energy function [Felzenszwalb and Huttenlocher, 2006],

$$E(f) = \sum_{p \in \mathcal{P}} \phi_d(f_p) + w_{sm} \cdot \sum_{p,q \in \mathcal{N}} \phi_{sm}(f_p, f_q). \tag{4.1}$$

Here, the unary term ϕ_d is called the data term, and the binary term ϕ_{sm} is called the smoothness or similarity term. $\mathcal{N}$ is the set of all neighboring pixels p, q. The energy function of Equation 4.1 is formulated for a stereo pair of cameras. However, the data term ϕ_d can easily incorporate the case of multiple cameras. In this case, one of the cameras is chosen to be the reference, and all other cameras are compared to this reference. The data term ϕ_d is then taken as the sum of the similarity measure between pixel p in the reference camera image and corresponding pixels p'_i in each of the other camera images. The smoothness term ϕ_{sm} only applies to the reference camera image.

Global methods such as Graph Cuts [Boykov et al., 2001], or Belief Propagation [Felzenszwalb and Huttenlocher, 2006] can be used for obtaining a solution for Equation 4.1. The weight w_{sm} controls the amount of smoothness that is imposed on the solution, where a larger value imposes a larger penalty on discontinuities. If on the other hand $w_{sm} = 0$ then Equation 4.1 becomes strictly local, i.e. the final depth value is determined independent of the value at other pixels.

4.3 Sensor Fusion via Local Method

Our goal is to fuse the information from different modalities for computing depth maps for the high quality reference camera. We will first describe our approach for sensor fusion using a local method. Later in this chapter we will describe sensor fusion using global methods. We therefore assume for now that $w_{sm} = 0$, and hence Equation 4.1 contains only the data term.

The problem is to obtain a high quality depth map by fusing (a) multi-view stereo (MVS) data from the reference camera plus satellite cameras, with (b) low resolution depth data from the depth camera, and incorporate (c) the information from the thermal signal. Sensor fusion can be achieved by constructing a data term which consists of different terms for the various modalities. We first consider contributions from the Time-of-Flight depth camera, and the satellite cameras. We formulate the data term as:

$$\phi_d = (w_{st} \cdot \phi_{d,st} + w_{ToF} \cdot \phi_{d,ToF} + w_{re} \cdot \phi_{d,sat}). \tag{4.2}$$

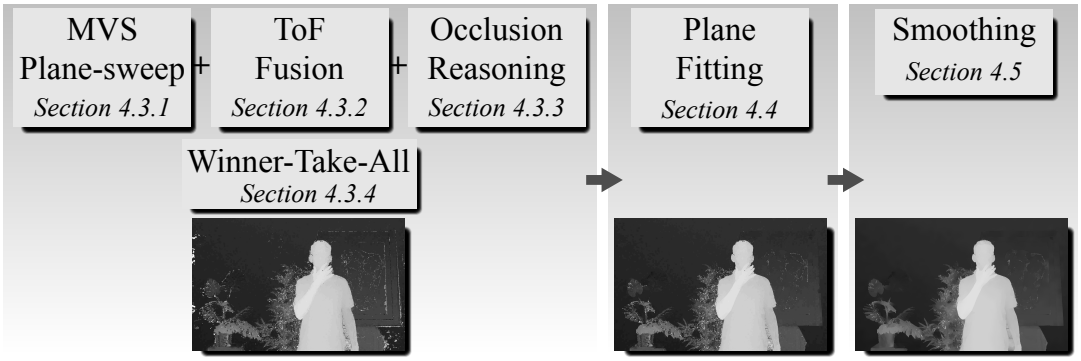


Figure 4.1: Overview. *Given data acquired with our experimental acquisition system from Chapter 3, depth maps are computed using different steps, illustrated in this figure.*

Here *st* stands for (photometric) stereo, *ToF* for Time-of-Flight, and *sat* for reprojection onto the satellite cameras with occlusion reasoning. The corresponding w parameters are weights to control the influence of each term. The resulting energy function becomes:

$$E = \sum_{\mathcal{P}} \phi_d. \tag{4.3}$$

We design the data terms in Equation 4.2 such that the solution for Equation 4.3 is found by selecting the depth with maximum support, i.e. largest value, for each pixel.

The depth computation using a local approach then contains the following steps (see Figure 4.1):

1. Compute the initial data cost $\phi_{d,st}$, for the multi-view stereo (MVS).
2. Fuse with Time-of-Flight depth data term, and data term for reprojection onto the satellite cameras.
3. Initial depth map determined by selecting depth with maximum support for each pixel.
4. Perform (conservative) plane fitting to improve initial depth map.
5. Smooth depth values using trilateral filter.

Next, we will explain the different data terms in Equation 4.2 in more detail.

4.3.1 Multi-View Stereo Depth using Satellite Cameras

We want to compute depth for the reference camera in our system. We do not rectify the images to compute disparities, but we compute depth values directly instead. Based on the depth layout in the scene, we choose a depth range and discretize this range into a set of depth layers. Each depth layer is parallel with the reference camera image plane. Given depth layer j with associated depth Z_j, and pixel i for the reference camera with image coordinates $(x_i, y_i, 1)$. The associated 3D point $\mathbf{X}_{ref} = (X_i, Y_i, Z_j)$ can be computed using $\mathbf{x} \cong \mathbf{K_{ref}}[I|0]\mathbf{X}_{ref}$ (the reference camera has identity rotation matrix, and zero translation):

$$\begin{aligned} X_i &= \frac{(p_x - x)Z_j}{f_x} \\ Y_i &= \frac{(p_y - y)Z_j}{f_y} \end{aligned}$$

The corresponding pixel on the satellite image plane is then computed as $\mathbf{x}' \cong \mathbf{K_{sat}}[\mathbf{R}|\mathbf{T}]\mathbf{X}_{ref}$.

Since all pixels are computed via a (3D) plane, we can omit explicit reprojection and instead warp the satellite images to the reference image. For a given camera with $\mathbf{K}, \mathbf{R}, \mathbf{T}$, from the 3D plane $z = Z_j$ we can infer the homography $H = \mathbf{K}[r_1 r_2 r_3 \cdot Z_j + \mathbf{T}]$, where r_i denotes column i of $\mathbf{R}$. Homography H relates points on the 3D plane to pixels on the image plane. We compute both H_{ref}, and H_{sat}, and then warp the satellite image to the reference image using $H_{sat}^{-1} H_{ref}$. Figure 4.2 illustrates the concept. Warped images are superimposed on the right for two depth planes, $\mathrm{D_A}$ and $\mathrm{D_B}$. As can be observed, the corresponding depths in the scene are "in focus". We can leverage the GPU to perform fast warping [Yang and Pollefeys, 2003].

Different color (dis)similarities can be used to characterize the difference between the colors of a pixel in the reference image and a corresponding pixel in the satellite image. Examples are sum of absolute differences, and Birchfield-Tomasi sampling insensitive dissimilarity measure [Birchfield and Tomasi, 1998]. The latter would be suitable in our case, since the reference and satellite cameras have different resolutions and field of views. However, although we apply color transforms to obtain similar color responses among the heterogeneous set of cameras, some residual color difference error remains. Normalized cross-correlation (NCC) is a more robust measure in the presence of such residual errors, since values are averaged over a (small) window:

$$NCC = \frac{\sum_{x,y}(f(x,y) - \bar{f})(f'(x,y) - \bar{f}')}{\sqrt{\sum_{x,y}(f(x,y) - \bar{f})^2 \sum_{x,y}(f'(x,y) - \bar{f}')^2}} \tag{4.4}$$

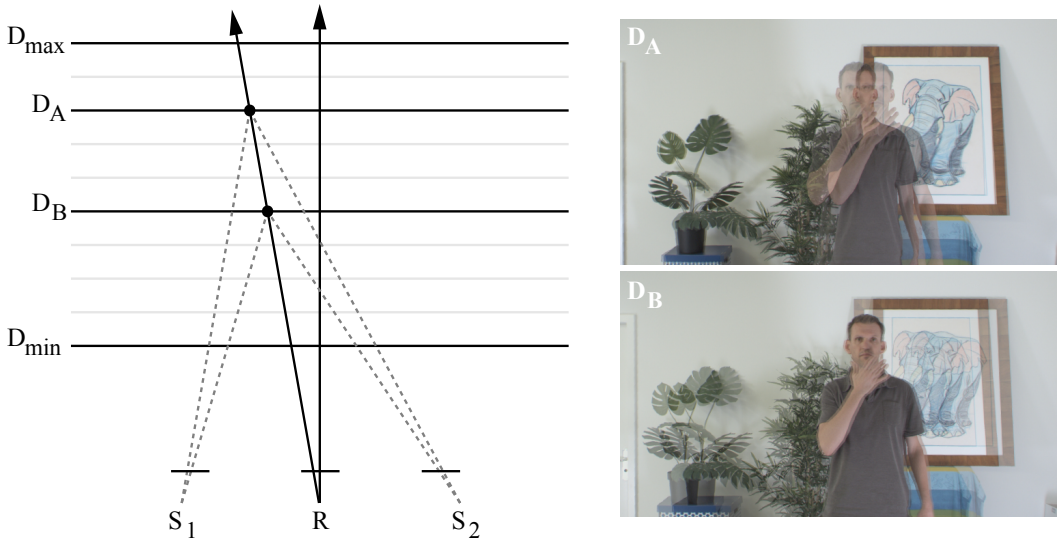


Figure 4.2: Discrete Depth Layers. *Reference camera R and two satellite cameras, S_1 and S_2. **Left** A depth range $[D_{min}, D_{max}]$ is discretized into a set of depth planes for reference R. **Right** Two examples of warped and superimposed satellite camera images corresponding to depth planes D_A and D_B. Objects located at the depth of the corresponding plane align in the warped images, and appear "in focus". As can be observed, depth plane D_A corresponds to the depth of the painting, whereas D_B corresponds to the depth of the persons' face.*

NCC gives values in the range between $[-1, 1]$, where a value of 1 means best correlation.

We adopt the adaptive NCC proposed by Heo et al. [2011]. To improve the localization power of the NCC a bilateral filter is incorporated. The bilateral filter combines a spatial Gaussian with a range Gaussian, based on the intensity. The bilateral filter weights are computed as:

$$w_{bil}^{pq} = G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|). \tag{4.5}$$

Here G_s is the spatial Gaussian, G_r the range Gaussian, p is the center pixel of filter window W, and $q \in W$. Pixel p has intensity I_p, and pixel q has intensity I_q.

The bilateral filter weights are applied to the correlation window:

$$NCC = \frac{\sum_{x,y} w_{bil}(f(x,y) - \bar{f}_{bil}) w'_{bil}(f'(x,y) - \bar{f'}_{bil})}{\sqrt{\sum_{x,y}(w_{bil}(f(x,y) - \bar{f}_{bil}))^2 \sum_{x,y}(w'_{bil}(f'(x,y) - \bar{f'}_{bil}))^2}}. \tag{4.6}$$

The means $\bar{f}_{bil}$ and $\bar{f'}_{bil}$ are also computed for bilaterally weighted values in the window. The size of the correlation window can be larger which makes the matching more robust. However, the bilateral weighting still provides good localization of intensity discontinuities.

We model the data terms in Equation 4.2 as truncated exponentials, and each term has values in the range $[\tau_{st}, 1]$. For increased robustness, the values are truncated at τ_{st} [Scharstein and Szeliski, 2002]. Correlation values near zero already indicate poor color matching and we therefore take $NCC = \max(NCC, 0)$. We can write the equation for the photometric stereo data term as follows:

$$\phi_{d,st} = \max\left\{\exp\left(\frac{-|1-\max(NCC,0)|}{\sigma_{NCC}}\right), \tau_{st}\right\}. \tag{4.7}$$

4.3.2 Fusion with Time-of-Flight Depth

The Time-of-Flight depth camera reports a 3D location $\hat{\mathbf{X}}$ for every pixel of the cameras' image sensor. To fuse the values from the Time-of-Flight camera, at every inferred 3D location for a depth plane Z_j of the reference camera, we want to compare Z_j with the Z value reported by the Time-of-Flight camera. We thus need to look up the value at the corresponding pixel in the Time-of-Flight image.

In Section 3.4.1 we determined that a 3D point $\mathbf{X}_{ref}$ in the reference camera coordinate system corresponds to a 3D point $\mathbf{X}_{ToF}$ in the Time-of-Flight camera via $\mathbf{X}_{ref} = Tr_{ToF\to ref}(\mathbf{X}_{ToF})$. To determine the pixel in the Time-of-Flight camera for which to lookup the depth value, we transform $\mathbf{X}_{ref}$ to $\mathbf{X}_{ToF}$ with $Tr^{-1}_{ToF\to ref}$, and then project onto the image plane with $\mathbf{K}_{ToF}\mathbf{X}_{ToF}$. The measured 3D location $\hat{\mathbf{X}}_{ToF}$ is then transformed to $\hat{\mathbf{X}}_{ref}$ in the reference camera coordinate system via $Tr_{ToF\to ref}$.

The data term for the Time-of-Flight depth data is then formulated as

$$\phi_{d,ToF} = \exp\left(\frac{-min(|Z_{ref} - \hat{Z}_{ref}|, \tau_{ToF})}{\sigma_{ToF}}\right). \tag{4.8}$$

Same as for $\phi_{d,st}$, the exponential is truncated for robustness. The parameters τ and σ_{ToF} should depend on the accuracy of the Time-of-Flight camera. We will discuss the choice of parameters in more detail in Section 4.8.

The depth camera is a time-of-flight sensor that measures phase shifts of reflected modulated IR illumination. However, some materials in the scene do

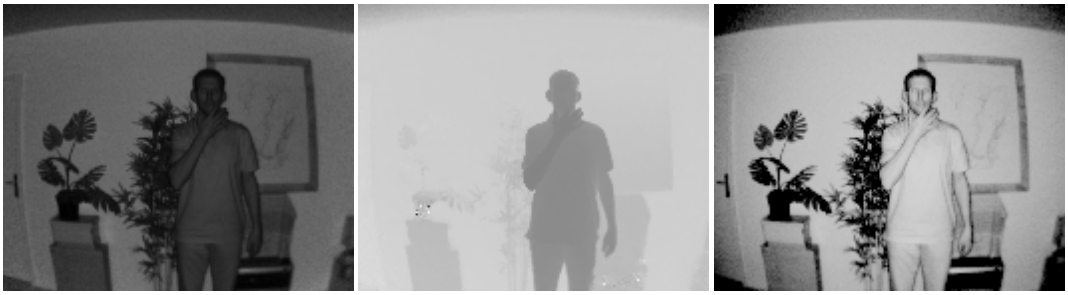


Figure 4.3: Time-of-Flight Camera Images. *Different images recorded by the Time-of-Flight camera.* ***Left*** *Grayscale image of reflected infrared signal.* ***Middle*** *Depth map.* ***Right*** *Confidence map indicating reliability of the measured depth values. Lighter values indicate higher confidence. Wrong depth values can be observed in regions where the confidence values are near zero.*

not reflect infrared light well. This affects the accuracy of the depth measurements in those regions. A *reliability* or *confidence* map could be computed based on the intensities in the acquired infrared image. However, for the specific hardware we use, such a confidence map is already constructed internally on the camera (see Figure 4.3). We exploit this confidence map to exclude low confidence areas in the fusion. From experiments we observe that for confidence values below threshold the depth measurements become unreliable. We therefore consider only depth measurements at pixels with confidence values above threshold.

4.3.3 Reprojection onto Satellite Cameras, with Occlusion Reasoning

The measurements from the Time-of-Flight camera may not correspond to the actual depth at that location. This is due to several reasons. One such reason is the case when an infrared light pulse strikes a surface at an oblique angle, for example along the depth silhouette of an object. The contribution recorded at the pixel may then come from light reflected by a surface located behind the object. Another reason is that the infrared light pulse received at a pixel may have traveled a so-called multi-path, i.e. the light bounces from one surface onto another, before being reflected towards the camera. In addition, the Time-of-Flight camera exhibits noise which may result in an incorrect measurement.

Fusion of incorrect Time-of-Flight measurement with stereo could thus lead to an incorrect result. The 3D location resulting from the fusion of stereo with Time-of-Flight depth can be additionally verified by comparing the color consistency between the reference camera and the satellite cameras. The basic premise is that if the 3D location is correct, the colors should agree. However, occlusions in the scene make this agreement harder to estimate in practice. Since our experimental acquisition system has two satellite cameras on either side of the reference camera, we can reason about possible occlusions. This is an important step to improve the overall quality of the depth map.

As explained in Section 4.3.1, a point $\mathbf{X}$ can be reprojected onto the satellite cameras image planes. We compute the absolute difference between the color associated with the pixel in the satellite camera, and the color of the pixel in the reference image. In the absence of occlusions we can compute ϕ_{re} as:

$$\phi_{d,re} = \exp\left(-\left(\frac{1}{n}\sum_{i=0}^{n}\left(\frac{\sum_c |I^c_{ref} - I^c_{sat_i}|}{3}\right)\right)/\sigma_{re}\right), \text{with } c = \{R, G, B\}. \quad (4.9)$$

Here, n is the number of cameras and I^c represents the intensity of a color channel.

Reasoning about Occlusions The absolute difference between the reference camera and a satellite camera i in Equation 4.9 can be denoted as $AD_i = \sum_c |I^c_{ref} - I^c_{sat_i}|/3$. For a given satellite camera, when the depth for a pixel under consideration is correct, we would expect that the color for the pixel in the satellite camera, matches the color for the pixel in the reference camera. Consequently, the corresponding AD would be small. By comparing the AD_i values for the satellite cameras, we can then reason about possible occlusions. Figure 4.4 depicts different occlusion cases for the symmetric satellite camera configuration of our experimental system. The satellite cameras are labeled 1 through 4, and the reference camera is labeled *Ref*. We additionally define $AD_{right} = \{AD_1, AD_2\}$ and $AD_{left} = \{AD_3, AD_4\}$. We can then distinguish the following cases for occlusion reasoning:

1. No occlusions:
   a) Correct depth: Equation 4.9 attains a minimum.
   b) Incorrect depth: AD_i for satellite cameras likely have different values and are larger compared to when the depth is correct.
2. Cameras on one side of the reference camera are occluded, depicted in Figure 4.4(a). We identify this case when:

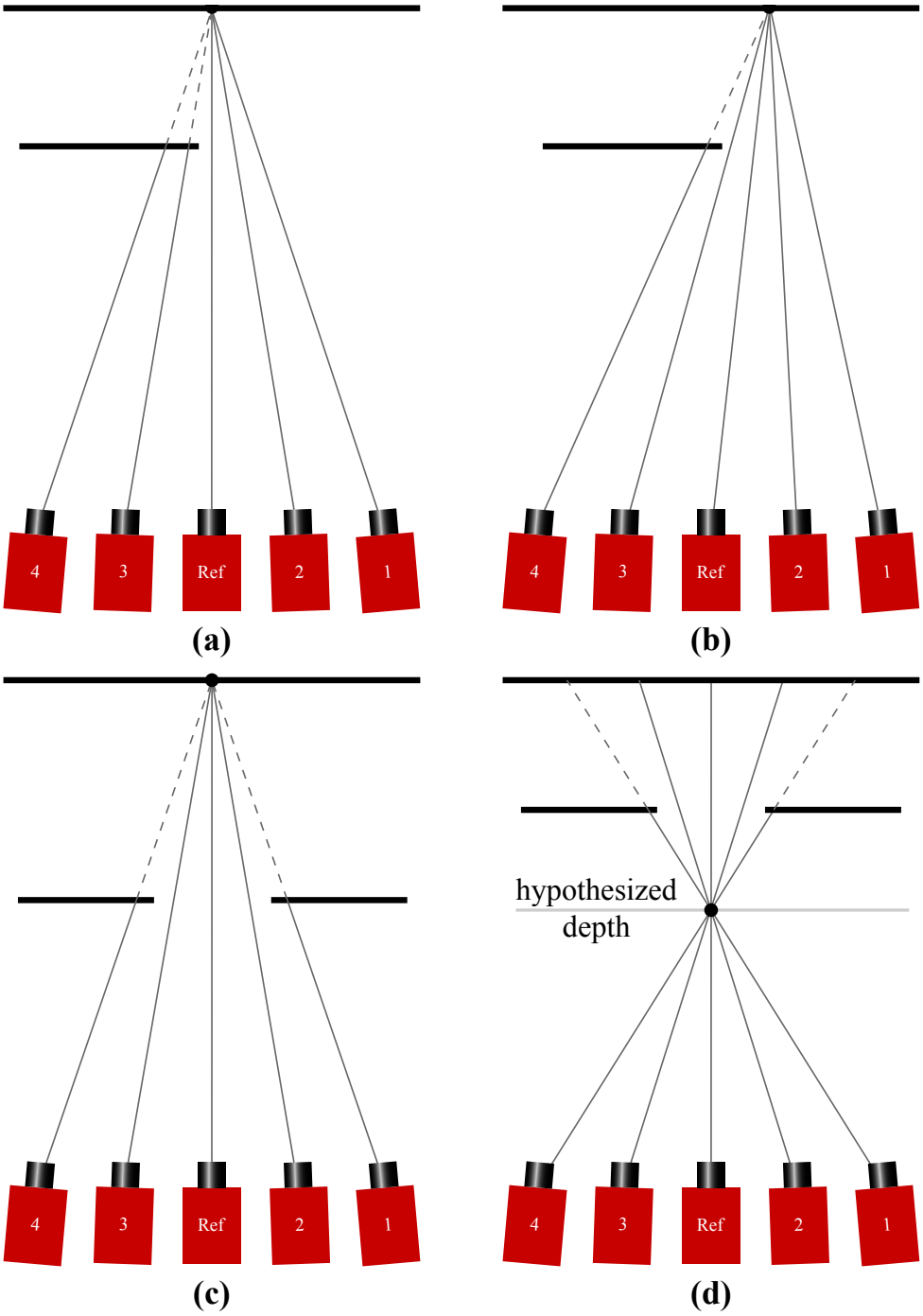


Figure 4.4: Occlusion Cases. *The reference camera is labeled* Ref*, and the satellite cameras are labeled 1 through 4. Heavy drawn lines represent surfaces at different depths. Camera rays are drawn to the point in depth under consideration.* ***(a)*** *Cameras 3 and 4 on the left are occluded.* ***(b)*** *Camera 4 is occluded.* ***(c)*** *Cameras 1 and 4 are occluded.* ***(d)*** *Although the hypothesized depth is closer than the occluding surfaces, this situation also flags cameras 1 and 4 as being occluded.*

   a) the values between AD_{right} and AD_{left} are different, and

   b) the absolute difference for AD_{right} are similar and below threshold, or

   c) the absolute difference for AD_{left} are similar and below threshold.

3. The outermost satellite camera is occluded, depicted in Figure 4.4(b). We identify this case either when:

   a) (camera 1 occluded) absolute difference for AD_2, AD_3 and AD_4 are similar, and AD_1 is large compared to those values, or

   b) (camera 4 occluded) absolute difference for AD_1, AD_2 and AD_3 are similar, and AD_4 is large compared to those values.

4. Both outermost satellite cameras are occluded, depicted in Figure 4.4(c) and (d): the absolute difference for AD_2 and AD_3 are similar.

5. The same as for the previous case, but now also one of the cameras adjacent to the reference camera is occluded.

For an occlusion occurrence of Case 2. we omit either cameras 1 and 2, or cameras 3 and 4 in Equation 4.9. For an occlusion occurrence of Case 3. we either omit satellite camera 1 or satellite camera 4 in Equation 4.9. For occlusion occurrence Case 4. we omit both cameras 1 and 4 from Equation 4.9, and finally for Case 5. we additionally omit either camera 2 or 3.

4.3.4 Winner Take All

We apply a Winner-Take-All (WTA) strategy to Equation 4.3, and assign depth values associated with the largest support value for each pixel. An example depth map is shown in the left column, top row of Figure 4.5. The depth volume for this example is discretized into fifty depth planes. The result shows that although WTA only considers each pixel independently of others, the quality of the depth map is good. In particular the depth values near the depth discontinuity of the foreground person are well resolved. Also the textureless wall is largely well reconstructed, which can be attributed to the fusion with the Time-of-Flight depth. However, the depth map exhibits noise which we aim to resolve next.



Figure 4.5: High Quality Depth Map. ***Left Column*** *Input data. From top to bottom: reference camera image, Time-of-Flight depth, and thermal image.* ***Right Column*** *Depth maps. From top to bottom: after initial fusion, after plane fitting, and after trilateral smoothing. After the initial Winner-Take-All fusion, the depth map is very noisy. Plane fitting reduces the noise, especially in planar areas. The final smoothing step produces high quality depth map, since depth boundaries are initially well estimated.*

4.4 Plane Fitting for Improving Depth Estimates

To improve the initial fusion result and reduce the noise we first segment the reference image into regions according to the photometric and thermal values of the pixels. We assume that the pixels belonging to a particular region

likely have a depth value which can be approximated by a plane. The goal is to determine a best fitting plane for a region, and if such a plane is found, estimate the depth values for all pixels in the region according to that plane. For segmenting the image into regions we adopt the superpixel segmentation proposed by Achanta et al. [2010]. We extend this method to incorporate the thermal signal in addition to the photometric information. By also considering the thermal signal we aim to correctly segment along object boundaries in areas with similar fore- and background colors. For each region we then perform the following steps:

1. Determine best-fit plane from depth values of four selected pixels.
2. Determine inliers and outliers for this plane hypothesis.
3. For the outliers, re-estimate the depths according to the plane hypothesis.
4. Reproject 3D points associated with the re-estimated depths onto the satellite cameras.
5. Determine if re-estimated depths are acceptable.

The above steps are performed according to RANSAC [Fischler and Bolles, 1981]. For each pixel we have the associated $(X, Y, D)^T$. Given four pixels selected for a region, we can fit a 3D plane $aX + bY + cD + d = 0$ using linear least squares. We discard regions for which we cannot estimate a plane with a sufficient number of inliers. Plane estimates which are too slanted with respect to the reference image are also discarded. For each plane estimate we compute the depths for the pixels classified as outliers according to the fitted plane. The corresponding 3D points are then reprojected onto the satellite camera image planes to check if the estimated depths are consistent with the color information from the satellite images (including occlusions, see Occlusion Reasoning below). If the depths are not consistent with the color information we discard the plane.

Our plane fitting approach is iterative, and only regions for which the pixels can be reliably estimated with a plane are processed on the first iteration. Reliable planes are those for which most of the pixels in the region are classified as inliers. For subsequent iterations we also consider regions which have similar photometric and thermal values compared to neighboring regions. We assume that these regions should have planes which are similar to estimated planes for the neighboring regions. We do allow the normals to differ within some threshold, which helps to better approximate regions that are curved.

If a scene is well approximated by planar segments, the plane fitting can reduce noise and achieve high quality results. In general however scenes may have detailed areas for which the depths are not well approximated by planes. Since our plane fitting is conservative this means that for some segments we cannot reliably estimate planes and we therefore leave the initial depth values unmodified. An example of the depth map after plane fitting is shown in the left column, middle row of Figure 4.5. Plane fitting improves the result in the background, and also for the foreground person. For regions of the plants, and also between foreground persons' body and arm, no planes could be reliably estimated. The final step discussed in Section 4.5 aims to improve those areas.

This plane fitting step can be considered an approximation to a global method, which imposes a smoothness constraint on the solution (Equation 4.1). By considering the depth values of pixels over regions, we essentially incorporate smoothness.

Occlusion Reasoning

We can take into account that we may be dealing with occlusions, and some pixels may project onto different surfaces in the different satellite images. We separately check for color consistency for the left and right satellite cameras. However, in addition we also check if the pixel in a satellite image is occluded by comparing the depth value. For each region which is well represented by a plane, we store the depth value at the reprojected locations in the satellite images. We overwrite the current depth value whenever a smaller depth value reprojects to a particular satellite image pixel. We can either iterate this approach for a fixed number of iterations, or until the number of segments that are updated is below threshold.

4.5 Smoothing

The depth values in the depth maps have been computed based on discretized depth planes. In addition we assumed that the depth values for pixels within a region can be approximated by a plane. Finally, as discussed above, some regions in the depth maps still exhibit noise in the depth values. The final step is therefore to perform a smoothing of the depth values. In edge-aware smoothing, intensity edges are respected to avoid smoothing across them. We adopt a similar approach and perform the smoothing of the

depth maps using a trilateral filter:

$$w_{pq} = G_{\sigma_s}(\|p - q\|)G_{\sigma_r}(|I_p - I_q|)G_{\sigma_d}(\|Z_p - Z_q\|). \tag{4.10}$$

Equation 4.10 extends the bilateral filter of Equation 4.5 with a term for the depth difference between pixel p and q. The trilateral filter therefore avoids smoothing over both spatial edges, as well as depth discontinuities. We extend the range component of the trilateral filter in Equation 4.10 to incorporate the thermal signal as well. In addition, we can also incorporate an additional term for the superpixel region boundaries, to avoid smoothing across neighboring regions which are dissimilar in color and thermal values. Trilateral smoothing is especially effective if the depth discontinuities have been accurately estimated [Smith et al., 2009]. The left-column, top-row of Figure 4.5 shows an example of the depth map after trilateral smoothing. The depth discontinuities which were already well estimated after plane fitting are preserved, while the noise in the depth values is smoothed. For the plant on the left-hand side the depth values are appropriately smoothed across the leaves.

4.6 Results for Fusion via Local Method

Additional results obtained with the local method we described are presented in Figures 4.6 and 4.7. Both Figures show the input image from the reference camera on the top row, the depth map after fusion but before smoothing on the middle row, and the depth map after smoothing on the bottom row. The depth volumes for these examples are again discretized into fifty depth layers. Textureless areas, such as the background wall, are difficult to reconstruct with only photometric stereo. Due to the fusion with the Time-of-Flight depth the depth values for these areas are reconstructed well in our local method. The fusion of the photometric information from the satellite cameras with the Time-of-Flight depth and incorporating the thermal signal, can also preserve detailed features such as fingers, even when the foreground and background colors are similar (Figure 4.6 right column). The depth for the interior part of the plant with the thin elongated leaves, on the right-hand side, is not reconstructed well. The fine details are not captured by the Time-of-Flight camera. For photometric stereo, pixels on leaves cannot be reliability matched due to the similarity and proximity of the many leaves.

By exploiting the thermal signal, the segmentation can be correct when there is no color discontinuity present. The left column of Figure 4.8 shows a zoomed-in area from the color image (for the example from the left column



Figure 4.6: Sensor Fusion using Local Method—I. ***First row*** *Reference camera color image (note: faces have been altered for privacy reasons).* ***Second row*** *Depth map before smoothing.* ***Third row*** *Depth map after trilateral smoothing. In the first column the hair of the two subjects is reconstructed correctly. In the second column the hand, including fingers, is accurately reconstructed.*

of Figure 4.6). Along the boundary between the subjects' hair there is no distinct color difference. The thermal signal in the middle column shows a clear difference, and the segmentation correctly segments along the boundary as shown in the right column. The depths are correctly reconstructed as a result (see Figure 4.6).



Figure 4.7: Sensor Fusion using Local Method—II. ***First row*** *Reference camera color image.* ***Second row*** *Depth map before smoothing.* ***Third row*** *Depth after trilateral smoothing. Additional examples showing results for challenging scenes including textureless areas, and cluttered background (plants).*

4.7 Fusion with Thermal Signal

The thermal signal is exploited during several of the steps to compute depth maps using a local method as explained earlier. Both superpixel segmentation and trilateral smoothing take the thermal signal into account. Since the thermal camera image is registered to the reference camera image, the thermal signal can be considered as an additional channel for the reference image. In other words, we can treat the combined reference image and thermal image as an (*R*,*G*,*B*,*Th*)-image.



Figure 4.8: Thermal Segmentation. ***Left*** *Region of interest in reference camera color image.* ***Middle*** *Region of interest in thermal image.* ***Right*** *Same as Left image, now with superpixel segment boundaries superimposed. There is no clear color discontinuity in the region where the subjects' hair overlap. However, the thermal signal does show a clear difference in the same region. The superpixel segmentation is able to segment along the thermal boundary (inside red rectangle).*

For the local method we have taken $w_{sm} = 0$ in Equation 4.1. Global methods on the other hand take the smoothness term in Equation 4.1 into account, and therefore $w_{sm} \neq 0$. The smoothness term acts as a regularizer. From a probability theory point of view, the smoothness term can be considered a prior. The prior assumption is that neighboring pixels which have similar colors should likely have similar depth values. Therefore, the prior aims to penalize the assignment of different labels (depths) to neighboring pixels with similar colors. This prior can be included in Equation 4.1 as a per-pixel spatially varying weight: $w_{sm}(p)$, for pixels p.

To incorporate the thermal signal, the gradients of the thermal image are combined with the gradients of the color image to determine the spatially varying weights. Solving Equation 4.1 now requires a global method, which considers all the pairs of neighbor pixels over the image, to compute a solution. Methods based on message passing, such as Belief Propagation, are well-known approaches for solving Equation 4.1. The data term for the energy function is the same as in Equation 4.2. Different functions can be chosen for the smoothness term. For example, a standard Potts model [Felzenszwalb and Huttenlocher, 2006] penalizes the difference between the labels of neighboring pixels, regardless of the *magnitude* of the difference. Instead, one can penalize with a function which depends linearly or quadratically on the difference between labels. For improved robustness, the cost function is typically truncated. From experiments we found that in our case a truncated

linear cost function gives adequate results:

$$\phi_{sm}(p,q) = min(|Z_p - Z_q|, T_Z); \tag{4.11}$$

We next present results for incorporating the thermal signal.

4.8 Results

The results in this section were computed using Tree-reweighted message passing (TRW-S) [Kolmogorov, 2006] for solving Equation 4.1. Figure 4.9 compares the thermal signal contribution. The left image shows the result when the thermal signal is incorporated as a smoothness prior, the right image shows the result without. Without the thermal signal, there is no distinct color discontinuity in the region where the actors' hair overlap. As a result, wrong depth is assigned to certain areas. On the other hand, there is a distinct discontinuity in the thermal signal, and the depth map now correctly assigns different depths.

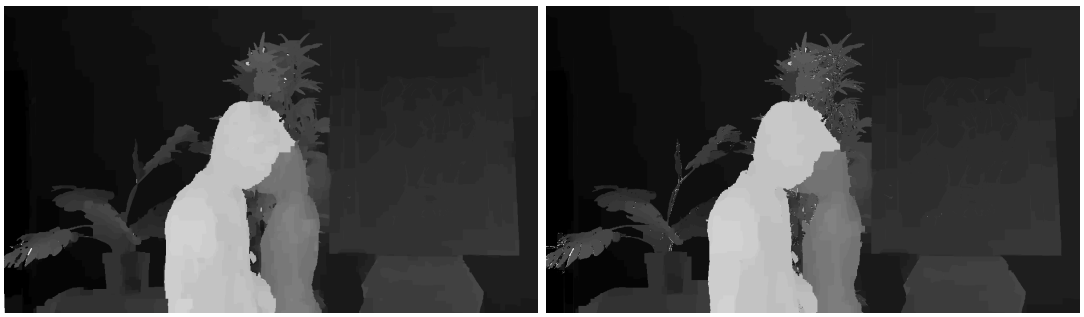


Figure 4.9: Depth Maps Computed using TRW-S. ***Left*** *With thermal segmentation prior.* ***Right*** *Without thermal segmentation prior. When not using the thermal signal in the smoothness prior, the boundary for the foreground objects' hair is incorrectly reconstructed, due to the lack of color gradient.*

There are several parameters in Equations 4.7, 4.8 and 4.9. From experiments we found that the following parameters values to be adequate: $\tau_{st} = 0.3$, $\sigma_{NCC} = 0.5$, $\tau_{ToF} = 0.1$, $\sigma_{ToF} = 0.14$, and $\sigma_{re} = 10.0$. The values for the Time of Flight camera are based on the reported accuracy of the camera. Our particular camera operates at $5m$, and in the case of near 100% reflectivity the camera accuracy is about $1cm$.

The weights in Equations 4.1 and 4.2 respectively determine how much smoothness to enforce, or how each modality is weighed relatively to the

others. In our results, each of the modalities is weighed equally and all three weights, w_{st}, w_{ToF}, w_{re} in 4.2 are set to 1.0. The weight w_{ToF} is set to zero when the value in the acquired confidence image falls below threshold, since the depth measurements for these areas are not reliable. The smoothness weight w_{sm} in Equation 4.1 is determined according to magnitude of the data cost. We validated experimentally that small variations of this weight have little impact on the final result. However, as expected, setting w_{sm} to larger values trades off the preservation of details with less noise in the depth map. An interesting direction for future research is to determine these weights automatically.

4.8.1 Comparison of Modalities

To give a better idea of the contribution of each modality for fusion, we omitted the reprojection onto the satellite cameras for the examples shown next. Figures 4.10 and 4.10 compare the results for stereo (RGB), stereo + thermal (RGB+T), stereo + depth (RGB+D) and stereo + depth + thermal (RGB+D+T), in rows two through five for several example scenes. The TRW-S parameters were fixed for all results. In the first column of Figure 4.10, (RGB) and (RGB+T) yield equivalent results. When the information from the depth camera is fused, the space between the two foreground subjects is reconstructed at the correct depth. Finally, in row four, (RGB+D+T) preserves the shape of the nose for the foremost subject. For the second column, when fusing the information from the depth camera, the paper leaflet is no longer being reconstructed compared to the (RGB) and (RGB+T) cases. This, together with the missing leaves for the plants in the background, demonstrates the problem of thin structures for the depth camera. In the third column of Figure 4.11, the hand shape is better preserved for both cases where thermal is considered in the smoothness prior. In the last column of Figure 4.11 the plant pot is reconstructed accurately along its boundary when thermal is considered.

4.9 Discussion

In this chapter we describe the computation of depth maps from data acquired with a reference camera augmented with satellite sensors. By fusing visible, Time of Flight depth and thermal information, we can employ a local method and achieve high quality depth maps for the reference camera. We compared our local method to a global message-passing method, and showed that we can obtain comparable results.

Experimental results were shown for scenes with dynamic objects and background clutter. Textureless areas, such as background walls, and also repeating texture patterns are handled by fusing depth from photometric stereo with the Time of Flight depth data. The Time of Flight depth sensor has a small resolution, and thin structures such as plant leafs are not adequately captured. Fusion with the photometric information from the satellite cameras allows us to reconstruct some of those details.

We also showed cases where surfaces of the same color overlapped at different depths. Of particular interest is the case where human subjects or body parts are overlapping —we showed that different subjects may have different thermal signatures, and therefore an occluding contour can be found even though no contours can be detected in the photometric information. We provided additional comparisons to show the contribution of each modality separately.

Although performance is not our main goal, it would be beneficial to be able to compute good quality depth maps at high performance. The performance of local methods is much higher compared to global methods. For the local method, results are computed in less than one minute per frame. The majority of the time is spent for the plane fitting and trilateral filtering. Both steps can be implemented on graphics hardware to increase the performance.

Our experimental system is only a proof of concept. Satellite sensors for this system are becoming more compact, more low-cost but higher quality. For example compact cameras with large imaging sensors are becoming available. Large imaging sensors exhibit less noise, and will benefit the quality of the depth maps. The resolution of most Time of Flight based depth sensors is their limiting factor. However, as these sensors are becoming more ubiquitous, combined with the recent introduction of depth sensors using structured light patterns, resolution is expected to increase. It is feasible to envisage a compact clip-on device that attaches to a high quality (cinematographic) reference camera, to enable robust and accurate computation of depth maps.

4.9.1 Temporal Consistency

The method presented in this chapter computes depth maps for each frame individually. This causes temporal inconsistencies in the computed depth maps. In the next chapter we discuss how we aim to address this issue. The insight we have is that consistent depth contours for foreground objects, typically human actors, is most important. Furthermore, depth contours are

correlated with color discontinuities. We thus propose to segment the foreground objects from the background, and in turn exploit the segment boundaries when computing depth maps. To ensure accurate segment boundaries, we propose an interactive method for generating the segment boundaries for the individual frames of a video sequence. We also discuss how we benefit from the multi-modal data we acquire with our experimental system. Two methods for computing depth maps using known foreground objects segment boundaries are discussed.

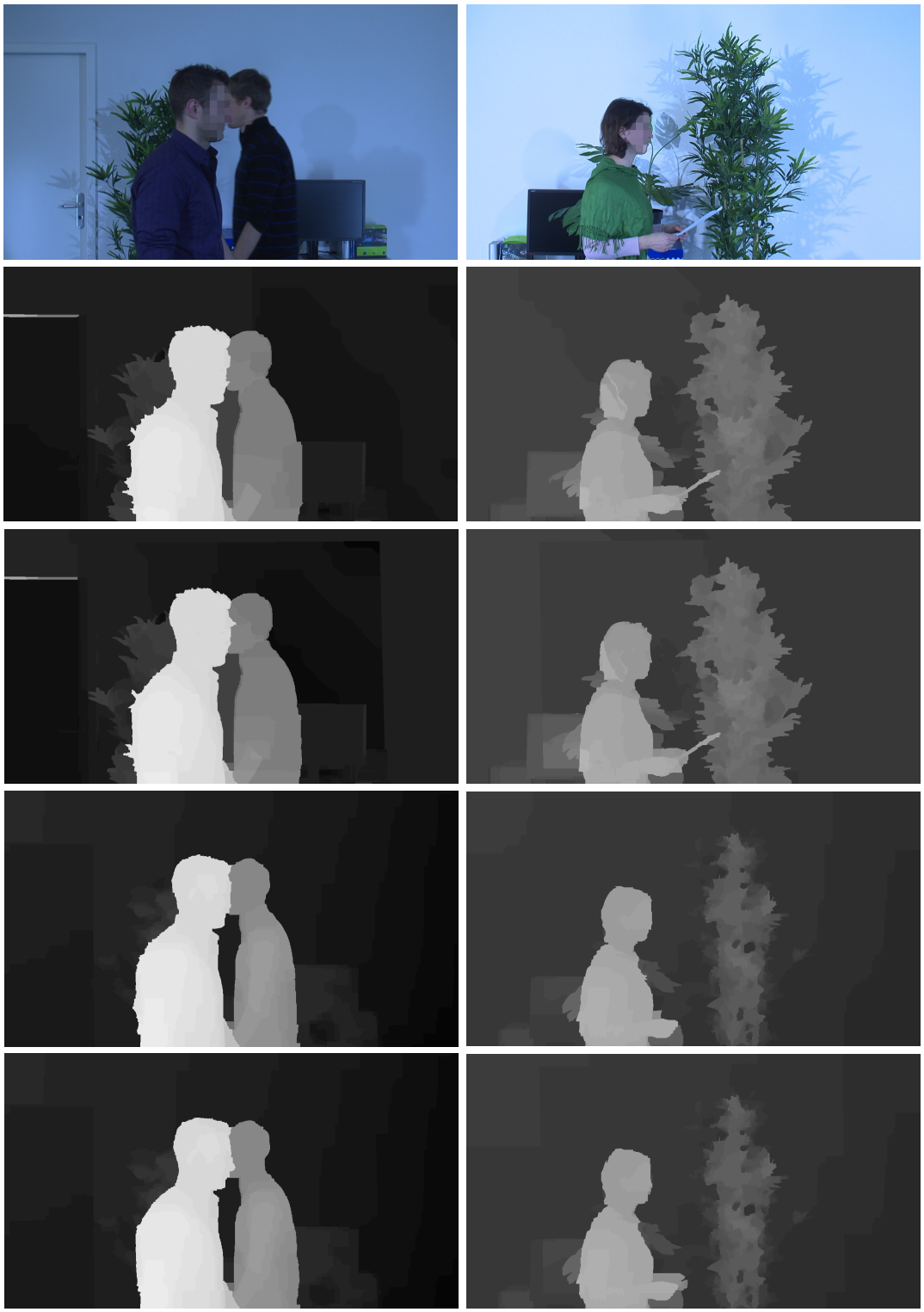


Figure 4.10: Several Example Scenes—I. *Comparison between stereo (2nd row), stereo + thermal (3rd row), stereo + depth (4th row), and finally stereo + depth + thermal (5th row). The first row shows the reference camera input images.*



Figure 4.11: Several Example Scenes—II. *Comparison between stereo (2nd row), stereo + thermal (3rd row), stereo + depth (4th row), and finally stereo + depth + thermal (5th row). The first row shows the reference camera input images.*

CHAPTER 5

Processing—Segmentation

Together with depth, segmentation is an important operation in the post-processing of acquired video. Segmentation involves the division of an image into regions belonging to the same object or background. Segment boundaries (for foreground objects) are usually object boundaries, and there is thus a correlation between segment boundaries and depth contours. In the previous chapter we described a method to compute depth maps from multi-modal data. Depth maps are computed on individual frames, without taking temporal information into account. Given the correlation between segment boundaries and depth contours, knowledge about accurate segment boundaries could thus be exploited for computing depth maps.

In this chapter we describe how accurate segment boundaries can be obtained for a video sequence. Our work is primarily focused on video sequences captured with the experimental system described in Chapter 3. Since our focus is on correctness, the method we present includes a user-in-the-loop to ensure correctness. We then describe how the segment boundaries are exploited for computing depth maps. Our results show that we can obtain high quality results, and with accurate boundaries that are temporally consistent, the depth contours are temporally consistent as well.

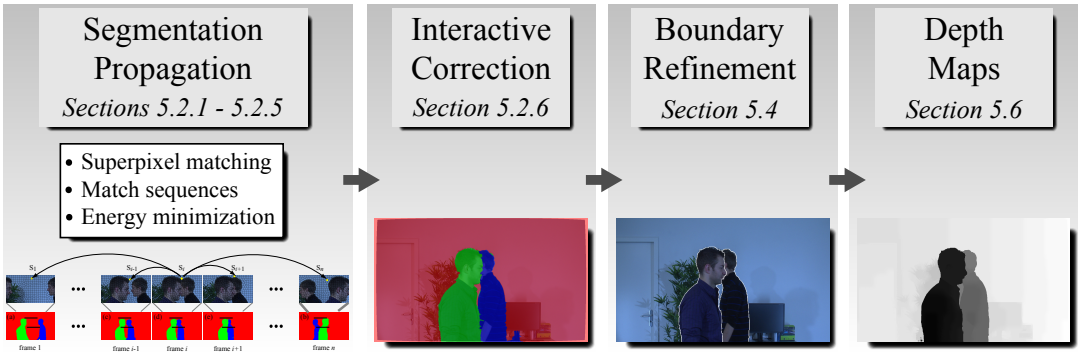


Figure 5.1: Overview. *Given data acquired with our experimental acquisition system from Chapter 3, the steps for propagating the segmentation and computation of depth are illustrated in this figure.*

5.1 Motivation

The term segmentation refers to the grouping of pixels which belong to the same object or region within an image. Different segmentation strategies have different goals. For example, over-segmentation, e.g. using mean-shift [Comaniciu and Meer, 2002], aims to ensure that no regions are merged, even when they belong to the same object. Other segmentation strategies actually merge the initial segmentation regions produced by over-segmentation, with the goal to cluster regions belonging to the same object.

Some segmentation strategies are entirely focused on the segmentation of a single foreground object, e.g. [Rother et al., 2004], while others focus on segmenting the image to classify regions, such as sky and grass [Kohli et al., 2009]. We are interested in the segmentation of one or more foreground objects from the background. Many movie shots contain human actors, and we are particularly interested in the segmentation of such scenes. Depth contours are correlated with segmentation boundaries. Typically a depth discontinuity has an associated color discontinuity. Accurate segment boundaries for objects in a video sequence can thus be exploited for computing depth maps. Furthermore, accurate segment boundaries across a video sequence will be temporally consistent, and by exploiting these boundaries the depth silhouettes would also be temporally consistent. This is important for subsequent editing based on the depth.

Segmentation relies mostly on color discrimination. Video segmentation additionally considers optical flow measures to produce temporally consistent segmentations, however optical flow itself relies on color discrimination as

well. Therefore, in the case when foreground objects, or foreground and background objects have similar colors, segmentation remains a challenging problem. Furthermore, consistent segmentation is challenging for occluding objects in a video sequence. We aim to overcome these problems by considering the segmentation of multiple objects across a video sequence as a labeling problem. In addition we can exploit our multi-modal data to make segmentation more robust.

For challenging cases, fully automated segmentation methods typically fail to correctly segment objects in the images. To meet the high quality requirements for cinema and broadcast, already many users are employed in interactive post-processing operations. Since our focus is on quality and correctness, we aim to keep a user in the loop to correct any segmentation mistakes. Rather than requiring tedious corrections at the pixel-level, for our method corrections are made on a more coarse level of pixel groupings, so-called superpixels.

In this chapter we describe two contributions. Our first contribution is an interactive segmentation approach. Given an initial segmentation for the first and last frame in a video sequence, our proposed method propagates the segmentations across the intermediate video frames. The problem is formulated as a labeling problem of smaller segments or superpixels, across the video sequence. Superpixels are matched to superpixels in adjacent images, without requiring the computation of optical flow, or camera motion. We will show that the labeling problem can be solved efficiently, while exploiting temporal coherence. By requiring an initial segmentation for the first and last frame of the video sequence, we can propagate the segmentation for occlusion occurrences in the scene. Finally, initial boundaries of the segmented objects are refined to obtain accurate object boundaries.

The second contribution in this chapter is to exploit the segmentation boundaries in computing depth maps. We describe this for two methods: an iterative non-global method, and a global method based on message passing. The iterative non-global method interpolates depth values up to the segmentation boundaries. This overcomes the problem that the Time-of-Flight depth does not accurately match the actual depth discontinuities in the scene, as explained in Section 3.2.3. For the global method we describe how to incorporate the segment boundaries into the message passing.

Figure 5.1 shows an overview of the steps involved for interactive segmentation, and depth map computations, together with the sections in which the corresponding parts are discussed in the remainder of this chapter.

Propagation Algorithm:

Require: Video sequence consisting of n frames: $\mathbf{I} = \{I_1, \cdots, I_n\}$,
and user provided segmentations for I_1, I_n.
Perform superpixel clustering on frames of $\mathbf{I}$.
for all intermediate frames I_i, $i \in [2, n-1]$ **do**
for all superpixels S_i^k **do**
Determine label from either I_1, or I_n.
Determine matches S_{i-1}^k and S_{i+1}^k.
end for
end for
Define matched sequences $\mathbf{S} = \{S_2^k, \cdots, S_{n-1}^k\}$.
Formulate as energy minimization problem.

Table 5.1: Propagation of Known Segmentations. *Outline of the algorithm for propagating segmentation labels over the frames of a video sequence.*

5.2 Interactive Segmentation

The goal is to segment frames in a video sequence into foreground objects and background, and obtain accurate segment boundaries. Fully automated segmentation methods may produce incorrect results for certain parts of the image. In offline applications it would be advantageous if a user could be incorporated into the process to ensure correct and accurate segmentation. The amount of work required by the user should be kept to a minimum. Ideally, most of the work is performed by an algorithm, and the user imposes high-level corrections when necessary. We develop an interactive segmentation approach with these thoughts in mind.

The outline of our segmentation algorithm is given in Table 5.1. Figure 5.2 depicts the procedure. Given a video sequence $\mathbf{I}$ consisting of n frames, a user provides the segmentation for the first and last frame. The goal is to propagate the segmentation to all intermediate frames of the video sequence. Propagation is achieved through matching so-called superpixels between adjacent frames. The problem of propagating known segmentations can then be considered a labeling problem, and formulated as an energy function. The propagation is then obtained by minimizing the energy function. We first introduce the energy function that we want to minimize, and then explain each term in more detail.

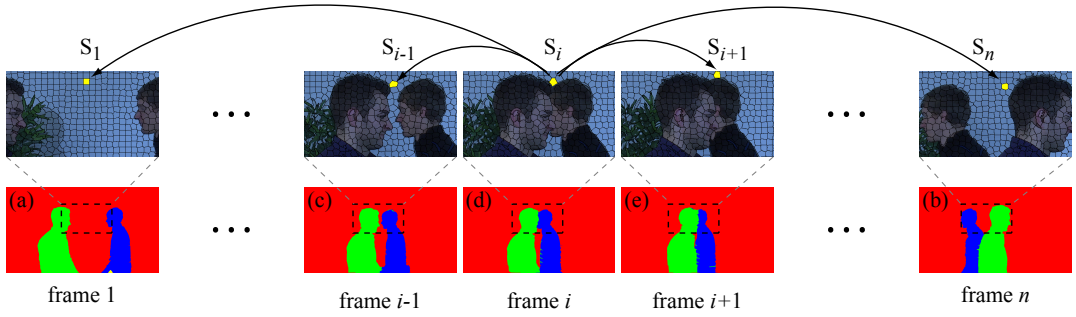


Figure 5.2: Segmentation Propagation. *A superpixel S_i in frame i is matched to superpixels in frames $1, i-1, i+1$, and n. Initial segmentations,* ***(a)*** *and* ***(b)****, are provided for frames 1 and n. The segmentation is then propagated to frames $2, \cdots, n-1$ by solving an energy minimization problem. The propagated segmentations for frames $i-1$, i, and $i+1$ are shown in* ***(c)****,* ***(d)****, and* ***(e)*** *respectively.*

5.2.1 Segmentation Propagation as Energy Minimization

We first perform a superpixel clustering[1] on each frame in the sequence $\mathbf{I}$. We choose the SLIC superpixels method [Achanta et al., 2010]. For SLIC superpixels, the user provides the desired number N_{sp} to obtain approximately equal-sized superpixels. The SLIC algorithm first divides the image into N_{sp} regular patches, with corresponding cluster centers. The algorithm then effectively performs an iterative K-means clustering using a combined CIELAB and spatial distance as its distance measure. The user can control the compactness of superpixels with a parameter which determines the degree by which the spatial distance is considered.

Pixels along color boundaries typically consist of a color mixture of the adjacent segments, and are therefore referred to as mixed pixels. These mixed pixels could result in elongated segments, or slivers. To avoid this, we first iteratively smooth the images by averaging with neighboring pixel groups in 3×3 blocks [Zitnick and Kang, 2007a]. We then perform SLIC superpixel segmentation on the smoothed images.

Segmentation can now be considered as the labeling of all superpixels over all frames in a video sequence. Given the user provided segmentations for frames I_1 and I_n, we formulate the segmentation propagation to in-between

[1] To avoid confusion we refer to over-segmentation into smaller regions as clustering, and to the labeling of foreground and background objects in an image as segmentation.

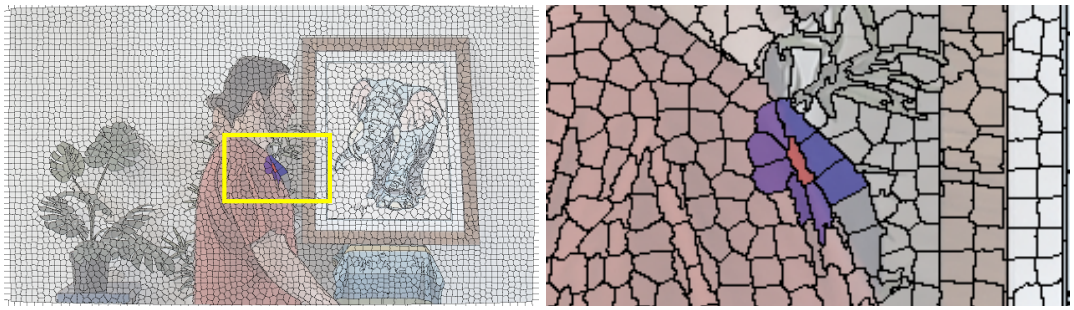


Figure 5.3: Superpixels. ***Left*** *Superpixel cluster boundaries superimposed on image.* ***Right*** *Close-up of area marked by yellow rectangle. The current superpixel under consideration is shaded red, and its set of neighboring superpixels are shaded blue.*

frames with the following energy function:

$$E = \sum_{i \in \mathcal{S}} \phi(x_i) + \sum_{(i,j) \in \mathcal{N}} \phi(x_i, x_j) + \sum_{c \in \mathbf{S}} \phi(\mathbf{x_c}). \tag{5.1}$$

Here, the unary term $\phi(x_i)$ represents the likelihood of a superpixel taking a particular label, with the set of labels $\mathcal{L}$ determined by the segments in the segmentation of I_1 and I_n. The binary term $\phi(x_i, x_j)$ represents the similarity between neighboring superpixels x_i and x_j in an image, and finally the higher order term $\phi(\mathbf{x})$ aims to enforce the same label for a collection of superpixels. We will explain each term in more detail next.

5.2.2 Matching Superpixels

To determine the cost of a superpixel taking on different segmentation labels, we determine for each superpixel S^k in the images $\{I_2, \cdots, I_{n-1}\}$ the best matching superpixel in either I_1 or I_n. For each superpixel we first construct a feature vector. Included in the feature vector is the average color over the superpixel, computed in the YC_bC_r color space. We found that this color space gave slightly better results compared to the RGB color space. We could have chosen the CIELAB color space instead, but we are not interested in the best perceptually matching superpixels. We also include the average thermal response for the superpixel if this modality is available.

We found that matching with average and thermal value alone did not give robust matching results, and superpixels in I_i are easily matched with wrong superpixels in I_1 or I_n. This is due to the fact that the averages are local measures for a superpixel. We therefore also incorporate the neighborhood for a

superpixel in the feature vector. An example of a superpixel and its neighborhood of superpixels is shown in Figure 5.3. We denote the neighborhood of superpixels connected to S^k as $S^{\mathcal{N}_k}$. We then compute histograms for Y, C_b, C_r, and thermal values over $S^{\mathcal{N}_k}$. Each histogram consists of 16 bins. In certain areas, e.g. near object boundaries, the set $S^{\mathcal{N}_k}$ may contain a number of different superpixels between subsequent frames. Although this could affect the matching quality, we found that matching was much more robust compared to including only information from a single superpixel.

The feature vector for a given superpixel is then given by:

$$f = (\bar{Y}, \bar{C_b}, \bar{C_r}, \bar{Th}, H_Y, H_{Cb}, H_{Cr}, H_{Th})^T. \tag{5.2}$$

Here the $\bar{(\cdot)}$ terms are the averages over the superpixel, and the H-terms are histograms. The cost of matching a superpixel S^k in I_i with S^m in either I_1 or I_n then becomes:

$$\begin{aligned} MC^{k,m}_{i\to\{1,n\}} = \|(&|Y^k_i - Y^m_{\{1,n\}}|, |Cb^k_i - Cb^m_{\{1,n\}}|, |Cr^k_i - Cr^m_{\{1,n\}}|, \\ &|Th^k_i - Th^m_{\{1,n\}}|, \chi^2_{dist}(H_{Y^k_i}, H_{Y^m_{\{1,n\}}}), \\ &\chi^2_{dist}(H_{Cb^k_i}, H_{Cb^m_{\{1,n\}}}), \chi^2_{dist}(H_{Cr^k_i}, H_{Cr^m_{\{1,n\}}}), \\ &\chi^2_{dist}(H_{Th^k_i}, H_{Th^m_{\{1,n\}}}))\|. \end{aligned} \tag{5.3}$$

Each term in Equation 5.3 is appropriately normalized. χ^2_{dist} denotes the χ^2-distance measure between two histograms H_i and H_j:

$$\chi^2(i,j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}. \tag{5.4}$$

To search for the matching superpixel in I_1 and I_n, we make no assumptions about the scene or camera motions. Figure 5.4 depicts the case where for superpixel S^k_i in I_i we search for matching superpixels in image I_1. We first determine the centroid of S^k_i. Then, we define a search radius r around the centroid location in I_1, shown by the shaded circular area in I_1 on the left-hand side of Figure 5.4. For each segment l within radius r the superpixel in I_1 with lowest matching cost is chosen, denoted as $S^{k,l}_1$. In the case of Figure 5.4 the search area intersects three segments, and hence we would obtain matches $\{S^{k,1}_1, S^{k,2}_1, S^{k,3}_1\}$ and corresponding matching costs $\{MC^{k,m}_{(i\to1),1}, MC^{k,m}_{(i\to1),2}, MC^{k,m}_{(i\to1),3}\}$. The procedure is repeated for the matching between images I_i and I_n, giving $\{S^{k,1}_n, S^{k,2}_n, S^{k,3}_n\}$ and $\{MC^{k,m}_{(i\to n),1}, MC^{k,m}_{(i\to n),2}, MC^{k,m}_{(i\to n),3}\}$. Note that we do not enforce uniqueness: superpixels in I_1 or I_n can be matched to multiple superpixels in I_i.

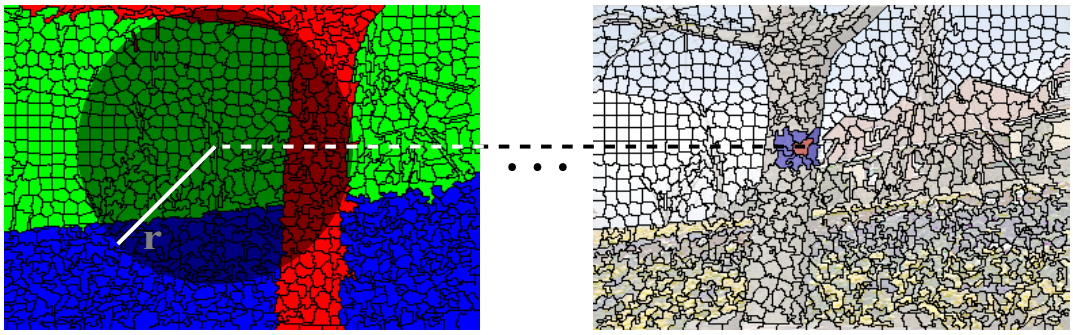


Figure 5.4: Matching Superpixels within Search Radius. ***Left*** *Image I_1 segmented into three segments: tree (red), sky + houses (green) and flowers (blue). Superpixel boundaries are superimposed.* ***Right*** *The current superpixel in image I_i (shaded red) and its neighborhood (shaded blue). The centroid of the current superpixel determines the centroid for a search area with radius r in image I_1. The superpixel within the search area with lowest matching cost is determined.*

A segment may not be matched in either I_1 or I_n. This could occur due to the object represented by a particular segment is not visible, or the segment lies beyond the search area determined by the search radius. In these cases a large cost γ is assigned for segment l.

$$\text{matching cost} = \begin{cases} MC^{k,m}_{(i\to\{1,n\}),l} & , l \in \mathcal{L} \text{ and } d <= r, \\ \gamma & , \text{otherwise.} \end{cases} \tag{5.5}$$

With $\mathcal{L}$ the set of labels, and d the distance between S^k and S^m. Finally, we choose the lowest matching cost between the matching costs for images I_1 and I_n: $\min(MC^{k,m}_{(i\to 1),l}, MC^{k,m}_{(i\to n),l})$.

This matching approach can handle non-rigid motions and moving cameras, at the expense of increased computational complexity. The user currently chooses radius r based on the motion in the scene. Large motions will require a larger search radius. We could incorporate a coarse estimate of optical flow to determine the search area. To increase the robustness of computing the matching cost, we take into account whether image I_i is "closer" in proximity to either I_1, or I_n. Proximity is measured according to the different search radii for images I_i. Given for image I_i the radii $r_{i\to 1}$ and $r_{i\to n}$, we weigh $MC^{k,m}_{(i\to 1),l}$ by $r_{i\to n}/r_{i\to 1}$, and weigh $MC^{k,m}_{(i\to n),l}$ by $r_{i\to 1}/r_{i\to n}$. As a consequence, for example for image I_2, the matching cost to I_1 will be weighed more compared to the matching cost to I_n.

Superpixel Matching with Adjacent Images

In addition to matching superpixels in I_i to superpixels in I_1 and I_n, we also match superpixels in I_i to superpixels in both adjacent images I_{i-1} and I_{i+1} for $i \in [2, .., n-1]$. For $i = 2$ we omit matching with I_{i-1}, and for $i = (n-1)$ we omit matching with I_{i+1}. In Section 5.2.5 we will explain in detail what these matches are used for. For this case we extend the feature vector in 5.2 with spatial location information:

$$f = (\bar{Y}, \bar{C_b}, \bar{C_r}, \bar{Th}, H_Y, H_{Cb}, H_{Cr}, H_{Th}, x_c, y_c)^T. \tag{5.6}$$

Here (x_c, y_c) is the centroid of the superpixel under consideration. The corresponding matching cost is now:

$$\begin{aligned} MC_{i\to j}^{k,m} = \|(&|Y_i^k - Y_j^m|, |Cb_i^k - Cb_j^m|, |Cr_i^k - Cr_j^m|, \\ &|Th_i^k - Th_j^m|, \chi^2_{dist}(H_{Y_i^k}, H_{Y_j^m}), \\ &\chi^2_{dist}(H_{Cb_i^k}, H_{Cb_j^m}), \chi^2_{dist}(H_{Cr_i^k}, H_{Cr_j^m}), \\ &\chi^2_{dist}(H_{Th_i^k}, H_{Th_i^m}), \|\mathbf{x}_i^k - \mathbf{x}_j^m\|)\|. \end{aligned} \tag{5.7}$$

The matching procedure is similar as described above. The search radius r is again chosen depending on the motion in the scene, however the radius is typically much smaller than for the case of matching superpixels between I_i and $I_{\{1,n\}}$. We do not weigh the matching costs by proximity in this case, and we do not compute per segment matches, but instead one match (S_i^k, S_{i-1}^m) and one (S_i^k, S_{i+1}^m). Again we do not enforce uniqueness, and we could have: $(S_i^k, S_{i+1}^m) \neq (S_{i+1}^k, S_i^m)$.

5.2.3 Incorporating Optical Flow

We can incorporate optical flow over the video sequence to improve the matching. We first compute the optical flow $F_{i\to j}$ between every pair of adjacent images (i, j) in the video sequence. We compute both directions $F_{i\to j}$, as well as $F_{j\to i}$. Then given an image i, for the matching of superpixels with frames 1 and n, we determine $F_{i\to 1}$ by compositing the flow computed for adjacent images:

$$F_{i\to 1} = \sum_{k=i}^{1} F_{k\to k-1}. \tag{5.8}$$

and same for $F_{1\to i}$. Next, for a superpixel S_i in image i we determine the matching superpixel S_1 in image 1, and vice versa for S_1 the matching superpixel S_i'. In the case when $S_i = S_i'$, the cost $MC_{(i\to 1),l}$ is assigned for the label l. The large cost γ to S_i is then assigned to all remaining labels in $\mathcal{L}$.

We similarly exploit optical flow information for matching superpixels between adjacent images (i, j). Using $F_{i\to j}$ and $F_{j\to i}$ to determine S_i and S_i'. If $S_i = S_i'$ the cost $MC_{(i\to j)}$ is assigned to the matching.

5.2.4 Initial Propagation

The binary term $\phi(x_i, x_j)$ in Equation 5.1 aims to enforce a first-order smoothness prior between neighboring superpixels in a frame, under the assumption that superpixels with similar color (and thermal signal) should likely have the same label. The binary term is defined by the following equation:

$$\phi(x_i, x_j) = \exp\left(\frac{(\mu_i - \mu_j)^2_{\{Y,C_b,C_r,th\}}}{\sigma^2_{\{Y,C_b,C_r,th\}}}\right). \tag{5.9}$$

Here μ is the average Y, C_b, C_r, th over the superpixel. The final value for $\phi(x_i, x_j)$ is computed as the average of the value for Y, C_b, C_r, th.

If we ignore the higher order term $\phi(\mathbf{x})$ for now, and taking only unary and binary terms into account, Equation 5.1 can be solved using Graph Cuts [Boykov et al., 2001]. The solution would assign a segment label to each superpixel in all in-between images $\{I_2, \cdots, I_{n-1}\}$. This results in a per-frame segmentation, without any temporal consistency between corresponding superpixels across frames.

5.2.5 Incorporating Temporal Information for Propagation

By using the higher-order terms, or clique potentials, $\phi(\mathbf{x})$ we aim to impose a temporal smoothness constraint on the superpixel labeling. Clique potentials penalize the assignment of different labels to some collection of variables, i.e. the clique [Kohli et al., 2009]. The penalty cost is irrespective of the number of variables that take on a different label:

$$\phi(\mathbf{x}) = \begin{cases} 0 & \text{, if } \forall x \in \mathbf{x}, x = l \\ \gamma & \text{, otherwise} \end{cases} \tag{5.10}$$

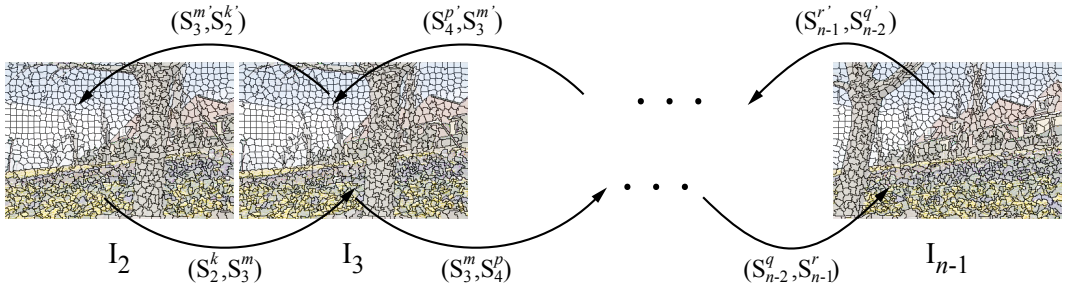


Figure 5.5: Superpixels Sequence. *Sequences of matching superpixels are constructed by considering pairs of matching superpixels for the images* $2, \cdots, (n-1)$. *Since it is not guaranteed that* $(S_i^k, S_{i+1}^m) = (S_{i+1}^k, S_i^m)$, *the process is repeated for images* $(n-1), \cdots, 2$, *to ensure each superpixel is contained in at least one match sequence.*

The robust extension to this allows some members of the clique to take on a different label [Kohli et al., 2009]. The robust clique potential is defined as:

$$\phi(\mathbf{x}) = \min\{\min_{l \in \mathcal{L}} \left(N \cdot \frac{\gamma_{max} - \gamma_l}{Q} + \gamma_l \right), \gamma_{max}\}. \tag{5.11}$$

Here γ_l is a per-label penalty, γ_{max} is the maximum penalty for the clique, and $N = |c_\mathbf{x}| - n_l(\mathbf{x})$, i.e. the number of variables in the clique which take a different label than l (where $|c_\mathbf{x}|$ is the cardinality of the clique). Q is called the truncation parameter and represents how many variables in the clique are expected to have a different label. In addition, we have that $N \leq \lfloor \frac{Q}{2} \rfloor$.

For our problem of segmentation propagation, we define cliques as *sequences* of matching superpixel correspondences over the video sequence. We use the matches computed between adjacent images as described in Section 5.2.2. Starting with a matching superpixel pair between images 2 and 3, (S_2^k, S_3^m), we add the pair to the match sequence S_k. We then continue with the matching pair S_3^m, S_4^p, and add S_4^p to S_k. We continue until image $(n-1)$ and for matching pair $S_{(n-2)}^q, S_{(n-1)}^r$ we finally add $S_{(n-1)}^r$ to the sequence S_k. Figure 5.5 illustrates the process. We repeat this process for every superpixel in all frames $2, \cdots, n-1$. We additionally repeat this process going in the opposite "direction", i.e. $n-1, \cdots, 2$. All superpixels which have not been included in a match sequence so far are processed at this time. Note that a superpixel may be included by multiple sequences. All match sequences together then form $\mathbf{S}$.

For each sequence S_k in $\mathbf{S}$ we store two sequences of matching costs:

$$C^{S_k} = \{MC^{k,m}_{2,3}, \cdots, MC^{q,r}_{n-2,n-1}\}$$
$$C^{S_k}_{\mathcal{L}} = \{MC^{k}_{2\to\{1,n\},l}, MC^{m}_{3\to\{1,n\},l}, \cdots, MC^{r}_{(n-1)\to\{1,n\},l}\}.$$

C^{S_k} stores the matching cost of superpixels between adjacent frames, and $C^{S_k}_{\mathcal{L}}$ stores the matching cost of superpixels with the segment labels (from either I_1 or I_n). From C^{S_k} and $C^{S_k}_{\mathcal{L}}$ we can determine the different terms in Equation 5.11. We first compute the mean μ_l over $(C^{S_k}_l)$ for each l in $\mathcal{L}$. We then define $\gamma_{l_{best}} = \min_l(\mu_l)$. The value of γ_l is then determined by:

$$\gamma_l = \begin{cases} \gamma_{l_{best}}, & \text{if } l = l_{best}, \\ \gamma_{max} = \gamma_{l_{best}} + \varepsilon, & \text{if } l \neq l_{best}. \end{cases} \tag{5.12}$$

Here ε represents a cost increase, and is discussed next.

Superpixel Sequence Matching Quality

Superpixels in a clique might have different segment labels. This is either due to incorrect matching, or due to an occlusion occurrence. We therefore want to allow that one or more of the superpixels in the clique will be assigned a different segment label. However, this should depend on the matching *quality* over a sequence S_k. The matching quality determines the truncation parameter Q (Equation 5.11) and cost increase ε (Equation 5.12). The expected segment labels for the sequence are determined by $S_{k,l_{init}} = \min_l(C^{S_k}_l)$. From this we can determine the *dominant* segment label as $N_{l_{max}} = \max_l(|S_{k,l_{init}=l}|)$. We can then determine truncation parameter Q as:

$$Q = \begin{cases} \frac{|\mathbf{x}|}{2}, & \forall l \in \mathcal{L}{:}\mu_l \text{ are similar}, \\ \min\left(\frac{|\mathbf{x}|-N_{l_{max}}+1}{2}, \lfloor\frac{|\mathbf{x}|}{2}\rfloor\right), & \text{otherwise} \end{cases} \tag{5.13}$$

For a good matching quality we expect the standard deviation over C^{S_k} to be small. To compute ε, we first compute $\triangle_{max} = \max(|C^{S_k} - \mu(C^{S_k})|)$, and then $w = 1 - \exp\left(-(\mu(C^{S_k}) + \triangle_{max})/\sigma\right)$. Finally, we set $\varepsilon = \gamma_{l_{best}} - w \cdot \gamma_{l_{best}}$. This allows some variables in the clique to have a different segment label with only moderate cost increase.

If a cost c_i in C^{S_k} is above some threshold, the corresponding sequence S_k may be split up into S'_k, S''_k. Each subsequence is then added separately to $\mathcal{S}$. We found that this improves the segmentation propagation result. Finally, Equation 5.1 is solved using Robust Graph Cuts [Kohli et al., 2009].

5.2.6 Interactive Segmentation Correction

Superpixels may be labeled incorrectly after propagation. In contrast to other approaches, incorrect labels may be easily corrected by the user. Corrections need to be made on the superpixel level, rather than requiring per-pixel corrections, and is therefore less tedious for a user. In the case where the propagation on a sequence fails and too many superpixels are labeled incorrectly, the propagation could be performed iteratively. In that case, the initial propagated segmentation serves as a starting point. The video sequence is then split into smaller sequences, and the segmentation propagation is then applied to these smaller segments.

5.3 Segmentation Boundary Refinement

The segmentation boundaries are determined by the superpixel clustering boundaries. Superpixels may include pixels from the background, or from other foreground objects, and as a result superpixel boundaries may not accurately match object contours. We therefore employ a refinement step to obtain accurate boundaries. We employ a boundary refinement step based on local overlapping classifier windows along the boundaries [Bai et al., 2009].

In our case an image is not just segmented according to a binary classification of foreground and background, but we have to take multiple segments into account. The first step is to determine a set of overlapping classifier windows along each segments' boundary. Here we take into account that boundaries may be shared between multiple segments. We denote the total set of classifier windows as $\mathbf{w}$. For each $w_i \in \mathbf{w}$ we model the color distribution for each segment in w_i with a 3-component Gaussian Mixture Model (GMM). We also incorporate the thermal signal when computing the GMM. For a given pixel in the window the probability $p_l(x)$ of pixel x having segment label l is determined from the GMMs as:

$$p_l(x) = p(x|l)/(p(x|l) + \sum_{l' \in \mathcal{L}, l' \neq l} (p(x|l')). \tag{5.14}$$

A pixel x may be included in several classifier windows. The final $p_l(x)$ is then a weighted average of the $p_l^i(x)$. The weights are calculated according to the distance of x to the center $(x_c, y_c)_{w_i}$ of the corresponding window w_i. The final segment label assignment is then refined using Graph Cuts segmentation [Boykov et al., 2001]. The boundary refinement steps can be iterated to progressively improve the accuracy of refinement.

5.4 Exploiting Multiple Modalities

The segmentation propagation we describe can benefit from multi-modal data in several ways. The thermal signal is incorporated in superpixel segmentation (Section 5.2.1), in the matching of superpixels (Section 5.2.2, and in the boundary refinement (Section 5.3). In addition to the thermal signal, we also incorporate the depth we obtain from the Time-of-Flight camera. The Time-of-Flight depth is not reliable enough to incorporate for the matching between superpixels. Instead, we use the Time-of-Flight depth to merge superpixels if their depths are similar. Merging reduces the total number of superpixels, and therefore reduces the processing time and problem size for robust Graph Cuts optimization.

5.5 Results

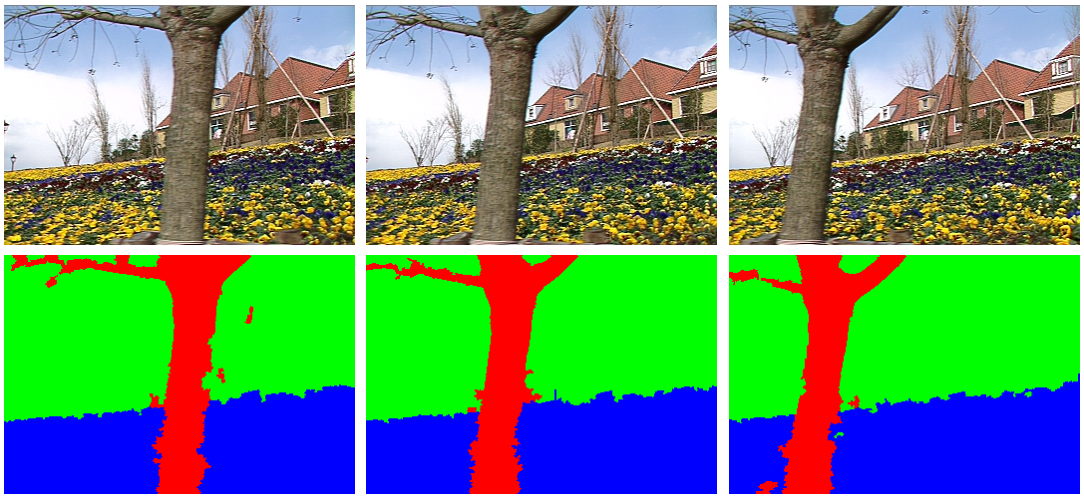


Figure 5.6: Flowergarden Result. *Result of our propagation method for frames 8, 17, and 28 of the flower garden dataset. The video sequence consists of 40 frames. The first and last image of the sequence have been segmented into three layers: tree, flowers, background. The results shown here are prior to interactive correction by the user, and shows the performance of our method on a standard dataset.*

Figure 5.6 shows the result of our propagation method for frames 8, 17, and 28 of the standard flower garden video sequence. We used 40 frames in this

example, and a three level segmentation was provided for the first and last frame of this video sequence. The propagated segmentations are prior to any interactive correction by the user, and prior to boundary refinement. Although some of the superpixels have been mislabeled, these results demonstrate segmentation propagation for an arbitrary video sequence.

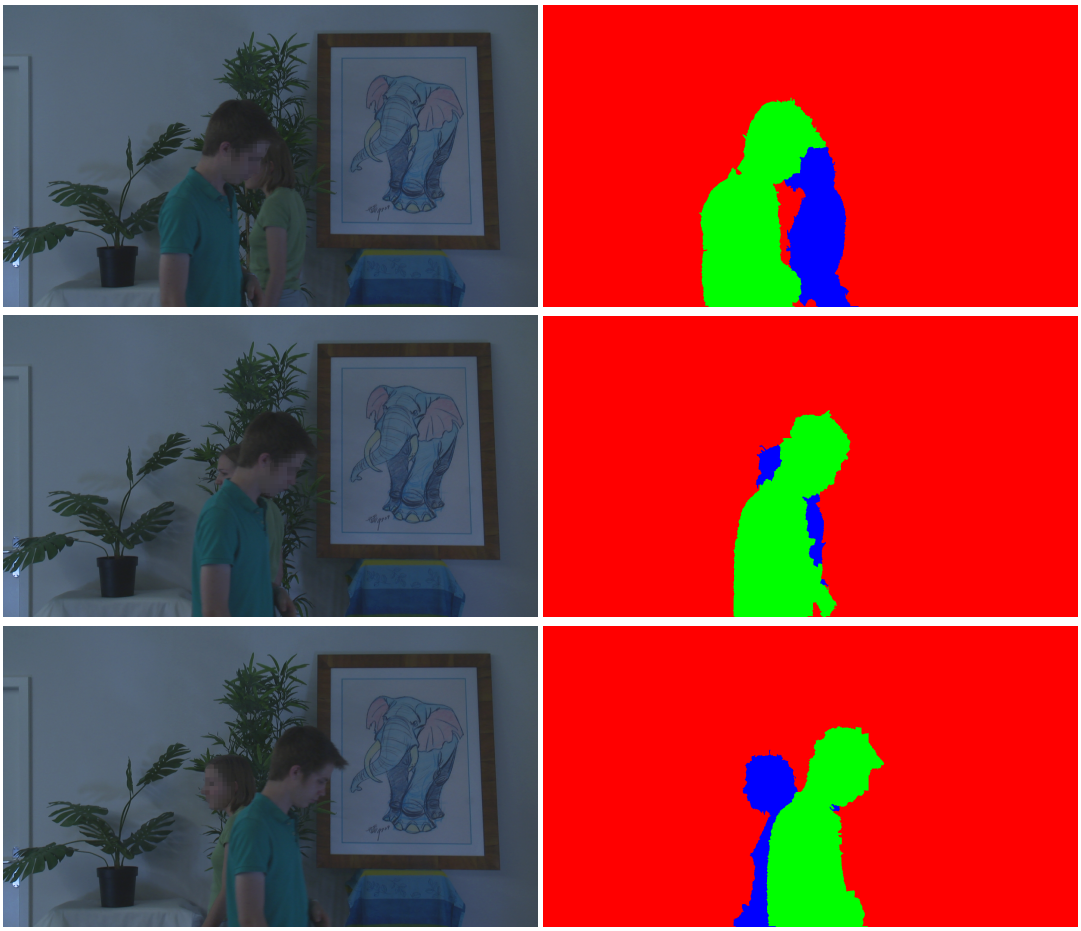


Figure 5.7: Occluding Objects Result. *Segmentation propagation for occluding objects.* ***Left*** *The input images.* ***Right*** *Segmentation propagation results prior to interactive correction and boundary refinement. Our method is able to propagate the segmentation through an occlusion.*

Figure 5.7 shows the result of a challenging case where one person occludes another as they walk past. This dataset was acquired with our prototype rig of Figure 3.2. For this example we incorporate motion information by using the optical flow method from Brox et al. [2004]. Figure 5.7 shows the results just before the occlusion (top row), during the occlusion (middle row), and just after the occlusion occurred (bottom row). The segmentation propagation results are before interactive correction and boundary refinement. By

exploiting the thermal signal, and defining clique potentials, the propagation can keep track of both people even though one person is nearly entirely occluded by the other. In particular, frames near the occlusion occurrence require interactive correction of the labels for a small number of superpixels, however this is crucial for accurate results.

Handling Occlusions

Occluding foreground objects with similar photometric properties are especially challenging for video segmentation methods. We require a known segmentation for the first and last frame of a sequence, such that we can handle occluding objects. The example in Figure 5.7 shows that combined with the higher order terms for the matching sequence, the propagation method assign the correct labels, with only few superpixels taking an incorrect label.

Interaction

After propagation some superpixels may have incorrect labels. A user can easily correct the incorrect labels. Figure 5.8 shows the result before and after interaction. On average the results require the user to correct around five superpixels per frame, with most of the correction required when similarly colored foreground objects occlude each other.

Boundary Refinement

The segment boundaries obtained from superpixels are refined using overlapping classifier windows as explained in Section 5.3. Figure 5.8 shows the result before and after refinement. Although the initial boundaries from the superpixel regions may be inaccurate, after refinement we get accurate boundaries.

5.6 Application: Depth Maps

5.6.1 Simplified Belief Propagation

This section describes the computation of depth maps for the reference camera using the segment boundaries. The computation is based on the simplified BP method in [Larsen et al., 2006], which is an iterative procedure relying



Figure 5.8: Boundary Refinement. *Segmentation boundaries before (**Left Column**) and after (**Right Column**) refinement. The refinement produces accurate boundaries even if the initial boundaries are inaccurate.*

on local neighborhood support rather than message passing. The computation exploits the segment boundaries computed in the previous section. An energy function, which is the same as Equation 4.1, is formulated for simplified BP:

$$E(f) = \sum_{p \in \mathcal{P}} \phi_d(f_p) + w_{sm} \cdot \sum_{p,q \in \mathcal{N}} \phi_{sm}(f_p, f_q). \tag{5.15}$$

The depth computation consists of the following steps:

1. Compute a distance map using the segment boundaries obtained as described previously.
2. Compute the initial data cost for the MVS by sweeping a depth plane through a discretized depth volume.

3. Perform an iteration of the simplified BP (utilizing the data from (1) as described below).
4. Recompute the data cost for the MVS given the current estimate of the depth map.
5. Iterate (3)-(4) until convergence.

Occlusion Reasoning: The computed depth map gets more accurate with each iteration, and this enables reasoning about occlusions in the satellite cameras. This supports a better estimate of ϕ_S because the computation can be done using only the cameras that see a particular point in a scene, without taking contributions from cameras that do not see the point due to occlusion. The following steps are used to determine the nearest object for each pixel in each satellite camera -

1. Iterate through each depth plane, starting with the farthest depth plane from the reference camera, and ending with the nearest depth plane to the reference camera.
2. Collapse all depths in the depth map which lie between the reference camera and the current depth onto the current depth plane.
3. Iterate through each satellite camera.
4. Find the mapping between pixels of the reference camera and the satellite camera, using the current depth plane.
5. Treat the collapsed depths as an image and warp them to the satellite camera, using the current depth plane (using the projective texture mapping capability of graphics hardware).
6. For each pixel in the reference camera, record the corresponding pixel in the satellite camera and the depth value from the warped depth image, where a depth value exists.
7. Move to the next depth plane and repeat (2)-(6).

The result is that for a given pixel in the reference camera and a given depth plane, one knows the corresponding pixel in each satellite camera plus the distance of that satellite pixel to its nearest object. If the distance to the depth plane is greater than the distance of the satellite pixel to its nearest object, then there is an occluder and information from the satellite camera is not used in the computation of ϕ_S.

Weights in the Energy Function: The ϕ_D term is weighted to reflect confidence in the depth value measured by the depth camera. The depth camera

is a time-of-flight sensor that measures phase shifts of reflected modulated IR illumination. It is less reliable near depth discontinuities where the IR signal strikes an oblique surface so that there is a reduction in the signal reflected back at the sensor. To take account of this, a distance map is computed based on the segment boundaries obtained in Section 4.3. The ϕ_D term has zero weight if the distance value is smaller than d_{min}, unit weight if the distance is greater than d_{max}, and a linear ramp in between. The ϕ_{sm} term is weighted to reflect confidence in the associated depth estimate. The goal is to propagate depth values from areas of high confidence to areas of low confidence as proposed in [Yang et al., 2006]. The confidence weight is given by $\phi_S(x_p) + \phi_D(x_p)$ with a weight of zero if this value is below a fixed threshold, and a linear ramp for values above the threshold. For the case in the previous paragraph where ϕ_D has zero weight, the confidence weight for ϕ_{sm} is also zero. In addition, ϕ_{sm} is only applied to pixels which lie within the same segment, using the segments obtained in Sections 4.2 and 4.3. This is analogous to previous approaches in BP where smoothing over photometrically dissimilar pixels is penalized.

Smoothing We apply the same trilateral smoothing as described in Section 4.5. Given the segmentation for an image, smoothing between pixels in different segments is disallowed.

5.6.2 Segment Boundaries for Message Passing

We also exploit the known segment boundaries in global methods based on message passing. A spatially varying smoothness term ϕ_{sm} in Equation 5.15 is again computed based on the gradient information from the reference and thermal image. The segment boundaries are then used to disallow messages between nodes from different segments. As depicted in Figure 5.9, a node p in the graphical model is connected to four neighboring pixels q_i. Node $q_1^{l'}$ belongs to segment l', whereas the remaining neighbor nodes belong to segment l. The messages between p and q_1 are disallowed.

5.7 Results

Figure 5.10 shows an example of the computed depth map using the segment boundaries. Although the hair and skin color are similar in some regions where the foreground objects overlap, the segment boundaries prevent the

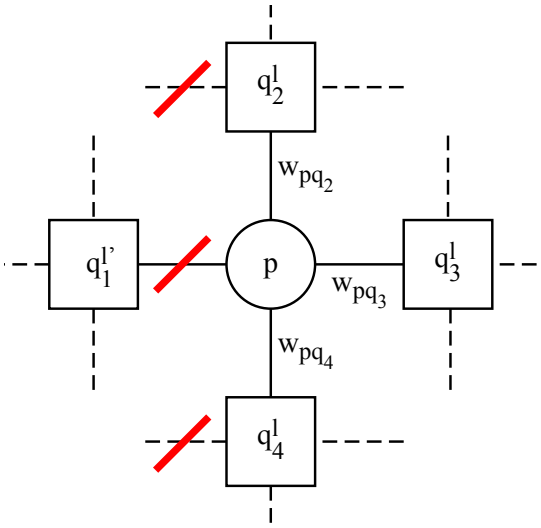


Figure 5.9: Exploiting Segment Boundaries. *Disallowing messages between neighboring nodes which belong to different segments. Node p belongs to segment l, whereas node q_1 belongs to segment l'. Messages between p and q_1 are disallowed.*

depth values from smoothing across depth contours of the objects. The examples of synthesized views using the smoothed depth map in Figure 5.10 show that the objects are well separated from each other and the background.

5.8 Discussion

In this chapter we describe an interactive video segmentation approach, with the goal to exploit the resulting segment boundaries in the images for computing depth maps. Video segmentation is formulated as a labeling problem over regions of pixels called superpixels. A segmentation for the first and last frame of a video sequence is required, and these known segmentations are propagated across the frames of the video sequence. Propagation is achieved through matching superpixels between images. By considering sequences of matching superpixels across the frames of the video sequence, we incorporate temporal information for the segmentation propagation. The corresponding energy minimization includes higher order terms for the match sequences, and can be efficiently solved using Robust Graph Cuts.

The matching of superpixels between images is straightforward, and since no assumption about motion is made, moving cameras and non-rigid objects can



Figure 5.10: Depth Boundaries. *Resulting depth maps and synthesized views, using segmentation constraints.* ***Top Left*** *Depth map before smoothing.* ***Top Right*** *Depth map after smoothing.* ***Bottom Left*** *Synthesized view using depth map before smoothing, no hole filling.* ***Bottom Right*** *Synthesized view using smoothed depth map. Segment constraints result in good depth discontinuities for the foreground objects. Segment constraints are also taken into account for smoothing of the depth maps, in order to maintain the good depth discontinuities.*

be handled by our approach. Furthermore, by propagating the information from both directions, our approach can handle occluding objects. The initial segment boundaries are finally refined to obtain accurate boundaries. For this, a set of overlapping classifier windows using Gaussian Mixture Models is used to determine per-pixel segmentation labels.

We can exploit the multi-modal data acquired with our experimental system. The thermal data is used for the matching of superpixels between images, to make the matching not only depend on photometric information alone. The Time-of-Flight depth may be exploited to determine if superpixels belong to the same object or surface. That is, if the depth between neighboring superpixels is similar, the superpixels may be merged together. This reduces the number of superpixels which have to be processed for a video sequence, and increase the stability of superpixels matching between images.

The accurate segment boundaries can be used as constraints when computing depth maps. We describe two methods which use the segment boundaries. One is an iterative method, which reprojects the depths onto the satellite camera in our experimental system. The other is a global method based on message passing. Results show that we can obtain depth maps with good depth contours. This is beneficial for post-processing operations such as view sythesis and insertion of Computer Graphs elements. In addition the segment boundaries are temporally consistent. The depth contours of the objects are therefore also temporally consistent. Since occlusions in depth maps occur around depth contours, being able to interpolate the depths up to the accurate contours results in high quality depth maps.

The method we present can be combined with the temporal filtering proposed by Lang et al. [2012]. Their temporal filtering aims to perform edge-aware smoothing of the depth maps, and may suffer from oversmoothing in the case when the foreground and background colors are similar. Incorporating the segmentation constraints proposed in this chapter could help resolve this problem, and provide temporal filtering of depth for both the foreground objects and the background.

Given the methods for computing segmentation and depth maps, in the next chapter we describe a method to copy and paste objects for stereoscopic 3D images. Stereoscopic 3D copy and paste requires both segmentation and depth maps.

CHAPTER 6

Processing—Stereoscopic Editing

There exist many *2D* image and video editing tools: the input is an image, or sequence of images, but the underlying scene depth is not considered. We would like to extend many of these tools to stereoscopic 3D. However, we cannot simply apply the 2D operations on the left and right images separately without taking the underlying 3D scene into account. Some, not all, 3D effects such as relative size depending on depth, or occlusions may be simulated in 2D images. Ensuring that this is done consistently for the left and right eye image is a tedious task. It would be desirable to have 3D operations which automatically take the underlying 3D scene into account, and consistently edit the left and right image of a stereoscopic 3D pair.

Thus far we have discussed methods for computing segmentation and depth for the images of a video sequence. In this chapter we discuss one of the more common editing operation of copy & paste applied to stereoscopic 3D images. Both segmentation and depth maps are required for stereoscopic 3D copy & paste. The goal is to copy one or more objects from a stereoscopic 3D source scene, and paste them into a stereoscopic 3D target scene. We ensure that objects change size depending on their depth or distance from the camera, objects change pose according to the underlying 3D surface, but

also adjust to the possible change in camera baseline, i.e. separation between the left and right camera, for the target scene.

6.1 Introduction

The system and methods discussed in Chapters 3, 4 and 5 are related to acquisition, segmentation and computation of high quality depth maps. We would like to exploit this information for additional editing of the stereoscopic 3D content. We cannot simply apply editing tools for 2D images or video, since editing for stereoscopic 3D content has to take the underlying 3D scene into account. Besides recovering the depth, editing tools also have to maintain comfortable stereo perception, including ensuring the correct handling of occlusions.

Our focus is mainly on live-action cinema and television. However, with the introduction of 3D TVs people can also view stereoscopic 3D at home, and even on mobile devices. In addition, consumer level 3D digital cameras [Fuji, 2009] enables easy capturing of 3D content. As with 2D images and video, there will be a need for editing this content. Given this, in this chapter we focus on a specific editing application: copy & paste for stereoscopic 3D images.

Copy & paste for 2D images has received a lot of attention in recent years [Pérez et al., 2003, Georgiev, 2006, Farbman et al., 2009]. The users' task for a plausible selection is to find objects which match in scale and orientation with that of the target. Objects can then be selected with a "rough" selection. No accurate segmentation of the object is required, provided that backgrounds are either uniformly colored or have similar texture. Simply applying these 2D methods in the source and target to the left and right eye images is not sufficient, since 3D copy & paste has to take stereopsis into account and avoid *stereopsis rivalry*: conflicting cues to the human visual system in the left and right eye images which could severely strain the visual system, or even destroy the 3D *illusion* altogether [Howard and Rogers, 2002, Patterson, 2007, Lambooij et al., 2009]. More specifically, important aspects are:

- Occlusion, being an important depth cue, has to be handled correctly.
- Maintain the copied objects' *stereo volume*, i.e., the anisotropic parallax between pixels that belong to the object and provide the cues for its 3D shape. Loss of this information leads to the so-called "cardboarding" effect, where objects appear as flat planes in depth.

- The composition result should be consistent for both left and right eye images. The pasted object should assume the correct orientation depending on the surface orientation in the target, which varies with the desired location for pasting.
- The copied object disparities in the target should be such that the depth composition is correct with respect to the depth in the target.

To take these aspects into account for 3D copy & paste, introduces the problem of recovering the depth information. Many existing methods for two-view stereo have been presented to compute per-pixel disparities [Scharstein and Szeliski, 2002]. However, for input images of arbitrary scenes the computed disparities are often inaccurate.

Furthermore, another challenge is to seamlessly composite the copied selection into the target. The aforementioned 2D copy & paste methods may result in smearing artifacts in the case where the backgrounds are dissimilar in texture. Only composition using alpha mattes can seamlessly blend objects with dissimilar backgrounds [Wang and Cohen, 2008]. High quality alpha mattes will require accurate segmentation of the object to be copied and pasted. Finally, direct rendering methods, e.g., forward mapping or geometry mesh approximation, may result in artifacts in the case of inaccurate depth maps.

In this chapter we discuss an end-to-end system for 3D copy & paste, consisting of components for depth reconstruction, selection and composition. Our system makes several contributions. Selection requires segmentation of the object(s) that will be copied. The first constribution is an automatic transfer of the segmentation from the left eye to the right eye image, which exploits the computed depth map. The second contribution is the registration of the copied object with respect to the local underlying support surface in the target scene. The third contribution is composition using so-called stereo billboards, which aim to preserve the original stereo volume of the source selection to prevent the object from appearing as a flat cut-out ("cardboarding" effect). The fourth contribution is the generation of contact shadows by transferring the disparity map to the target and using an image space ambient occlusion approach.

6.2 Stereoscopic Copy & Paste

Our 3D copy & paste system allows a user to select objects from several stereoscopic source images and composite them into a desired stereoscopic target image. An overview of our system and the editing workflow for 3D

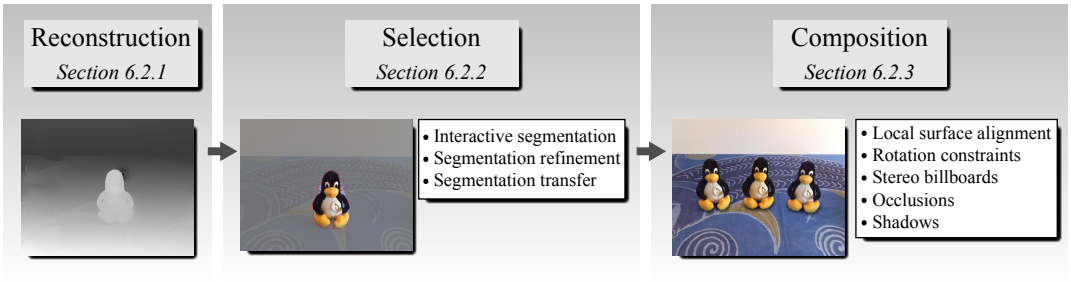


Figure 6.1: Overview. *The system for stereoscopic copy and paste consists of three components, illustrated in the figure.*

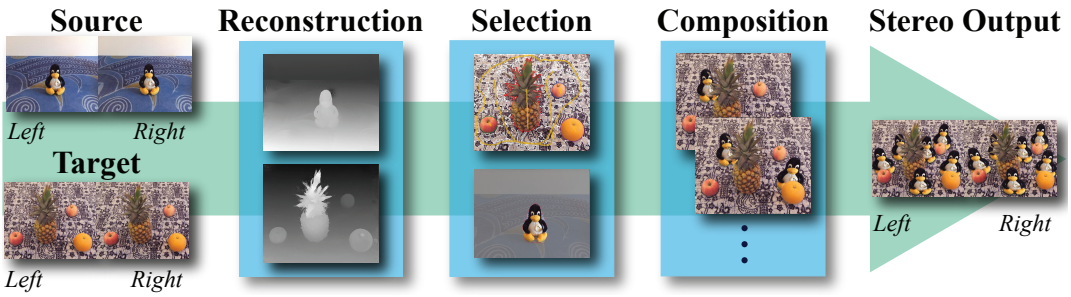


Figure 6.2: Workflow for 3D Copy & Paste. *Given a stereoscopic pair of source and target images, the first component is depth reconstruction, which could be performed offline prior to online editing. Next the user performs segmentation and selection of the object(s) to be copied. Finally the copied object(s) is pasted into the target at some desired location, and the result is a composited stereo pair of images.*

copy & paste are shown in Figure 6.1 and Figure 6.2 respectively. Input to the system are stereoscopic pairs of images for the source and target. The system can be divided into three components:

1. Depth Reconstruction.
2. Selection.
3. Composition.

The first component, Depth Reconstruction (Section 6.2.1), determines the underlying 3D scene for both source and target. The depth is used during selection to support segmentation transfer, and during composition to support object placement, occlusion handling, and the stereo billboard steps. The

main challenge for editing is in handling inaccuracies and wrong values in the computed depth maps.

In the next component, Selection (Section 6.2.2), the user selects one or more objects from source images to be copied to any desired location in the target. To support this goal, several steps are necessary in preparing the source and target images. Accurate boundary segmentation of objects, ground planes, backgrounds etc. in both source and target images is required. We have implemented an interactive segmentation tool. To reduce the amount of required user input and ensure consistent segmentations, the segmentation for the left eye is automatically transferred to the right eye image.

In the final component, Composition (Section 6.2.3), the user determines a desired location for pasting the copied selection in the target. Composition is performed interactively while the user is viewing the resulting composite stereoscopically. The system continuously ensures consistent orientation of the cloned object with the local orientation in the target, by computing a best-fit alignment with the targets' local underlying surface. Furthermore, since only two views are available and to avoid the need for in-painting, the system constrains the amount of rotation and aims to keep the objects "forward facing". To ensure that the stereo volume of the objects is preserved, and avoid the cloned objects from appearing flat, we have developed a method we refer to as stereo billboards. Copied objects are sorted in depth for correct occlusions. Finally, our system computes approximated contact shadows to avoid the copied objects from appearing to float. We will next describe the individual components in more detail.

6.2.1 Depth Reconstruction

We could use the methods described in Chapters 4 and 5 to compute the depth maps, in the case when the data is acquired with our experimental system. In the general case where only a stereoscopic pair of images is available, we instead use the method presented by Smith et al. [2009] to compute disparity maps. For each pair of images we compute the disparity map $D_{l\to r}$, between the left and right image, and $D_{r\to l}$ between the right and left. Since disparity and depth are related [Hartley and Zisserman, 2004, Ch. 10], we simply refer to disparity as depth instead.

In general, the depth values in certain areas may not correspond to the correct depth due to the limitations of the particular algorithm. Furthermore, depth values may be incorrect due to occlusions between the left and right

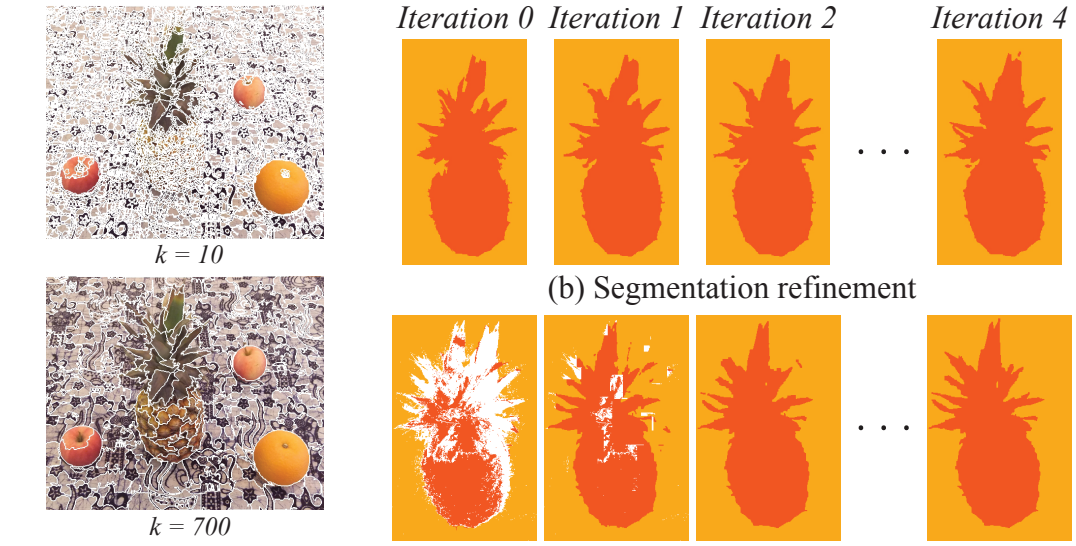


Figure 6.3: Object Selection. ***(a)*** *Adjusting the kernel size can ease multi-object segmentation, because the largest clusters usually correspond to separate objects.* ***(b)*** *Segmentation refinement of the pineapple through four iterations of graph cuts optimization.* ***(c)*** *Segmentation transfer results from the left eye image to the right eye image. Pixels with unknown segmentation are shown in white.*

eye images. Our system is thus designed to be able to perform copy & paste editing in the presence of (locally) inaccurate depth maps.

6.2.2 Selection

Our goal is to provide the user with the flexibility of selecting multiple objects from the source, and paste them at any desired location in the target. To support this goal, both source and target should be accurately partitioned into segments corresponding to objects, surfaces and backgrounds. Accurate real-world object segmentation requires a significant amount of user interaction in the form of strokes to mark fore- and background pixels. To reduce the amount of user interaction we have implemented an interactive multiple object segmentation approach with automatic transfer from one eye to the other.

Interactive segmentation Interactive segmentation refers to the segmentation of one or more foreground objects in an image. The method described can be used to provide segmentation for the first and last frame for the interactive segmentation propagation of Chapter 5. Segmentation starts by computing a mean-shift clustering [Comaniciu and Meer, 2002] on the image. This results in some initial segmentation of the image into fore- and background objects. Increasing the mean-shift kernel size, more aggressively merges clusters across the entire image. We observe that, compared to the background surfaces, the foreground objects' contours require a smaller mean-shift kernel size to better preserve the details. We thus employ the following scheme: the user adjusts the kernel size until the foreground objects are sufficiently clustered into an initial segmentation (Figure 6.3a), next the user provides strokes to merge clusters and improve the segmentation of the foreground objects. These two steps can be repeated until some desired segmentation of the foreground objects has been achieved. The remaining clusters of the background surface can then be merged with only a small number of strokes.

We again exploit the correlation between depth contours and color discontinuities, and incorporate depth as a fourth channel in the mean-shift to improve the cluster boundaries. Furthermore, the user can adjust the kernel size adaptively for each object.

We merge clusters using the maximal-similarity merging mechanism [Ning et al., 2010]. Clusters covered by the users' stroke are first merged and marked as selected, and the selection is then updated iteratively. More specifically, if cluster R is selected, we merge cluster Q with R if:

1. R and Q are adjacent, and
2. $\rho(R,Q) = \max_{S \in \mathcal{N}(Q)} \{\rho(Q,S)\}$.

Here $\mathcal{N}(Q)$ denotes the set of adjacent clusters to Q, and $\rho(R,Q)$ measures the similarity of two clusters for color and depth. Instant visual feedback is provided to the user during sketching, similar to Paint Selection [Liu et al., 2009b], allowing the user to decide whether to continue or stop sketching.

Segmentation refinement with localized classifiers User input strokes help differentiate objects in the scene. However, due to color ambiguity or estimation errors in the depth maps, the contours of the merged clusters may not fit the object boundary accurately (see Figure 6.3b—*Iteration 0*). Therefore, after each stroke sketch, the contours are refined by applying graph cuts optimization [Boykov et al., 2001] using overlapping localized classifiers [Bai

et al., 2009]. We use the same refinement after transferring the segmentation to the right image. We first discuss the contour refinement for the segmentation in the left image.

Bai et al. [2009] assume an accurate segmentation of the first frame as input. They then define a set of overlapping windows whose centers lie on the segmentation boundary. Each window contains both background pixels and foreground object pixels. Color statistics for each window are gathered, and a classifier assigns to every foreground pixel within that window, a probability of that pixel belonging to the foreground. Bai et al. advocate using small local windows. However, as stated above, our initial segmentation may be inaccurate and hence, the local statistics for small windows may be incorrect. Larger windows would then be required for the inaccurate areas along the boundary. Since there is no knowledge of where the inaccurate areas are, we create two different sized windows at each sampled location on the boundary: one small (30×30 pixels) and one larger (60×60 pixels). For each window we build a Gaussian Mixture Model (GMM) in the Luv color space using local color statistics. In addition, we use information from the whole image to build a global GMM. For each window size we then compute the model confidence for both local and global GMMs ([Bai et al., 2009, Eq. 2]), and we pick the one with the highest confidence. We run several iterations of 2-label graph cuts refinement for each input stroke. After each iteration we update the local classifiers along the new boundary. Refinement iterations are shown in Figure 6.3b.

Consistent segmentation transfer. To avoid the need for the user to repeat the segmentation procedure for the right image, we transfer the segmentation result from the left image. We exploit the depth map and only transfer those pixels with coherent disparities between the left and right images, since those pixels likely have classifiers with strong confidence. A pixel is said to have coherent disparities if:

$$|d_{l\to r} - d_{r\to l}| \leq 1 \tag{6.1}$$

The initial segmentation transfer result is shown in Figure 6.3c—*Iteration 0*. Since the image after transfer is initially sparsely segmented, we also transfer the local classifiers from the left image. However, we compute a new global GMM on the second image using only pixels with coherent disparities. We compute the confidence values as described above and pick the one with highest confidence. We perform several iterations of $k+1$-label graph cuts for global refinement for k partitioned segments. Several iterations are shown

in Figure 6.3c. If the automatic transfer does not give the desired quality of segmentation, the user may provide additional strokes for refinement.

6.2.3 Composition

In the final component of our system the user composites (pastes) the selection in the target images. To support interactive exploration of the location for pasting, we aim for interactive performance while the user observes the resulting composite in stereo 3D. However, as explained in Section 6.1, composition needs to take the various aspects related to stereopsis into account: target depth composition, consistency, occlusions, and stereo volume. To address these aspects we perform the following steps:

- Alignment of the pasted object with the local underlying surface in the target.
- Constraining the rotation of the pasted object to avoid the need for inpainting or object completion.
- Stereo volume preservation using stereo billboards.
- Depth sorting to determine the correct visibility, i.e., occlusions.
- Shadow estimation using the depth map and an ambient occlusion technique.

Inaccuracies in the depth maps preclude direct artifact free rendering of the selection, either using, for example, point sample rendering [Zwicker et al., 2002], or mesh fitting [Zitnick et al., 2004]. For robustness with respect to inaccuracies in the depth maps we introduce the stereoscopic extension of billboard rendering which we have labeled stereo billboards. In the remainder of this Section we will explain the above steps in more detail. For all our methods, we represent the geometry (point clouds) of both source and target scenes in a common coordinate frame. We define the center of projection of the left eye camera as the origin of a 3D coordinate system, and align the source and target camera to lie at the origin of this frame.

Local Surface Orientation Alignment

In the real world, objects are typically placed on some supporting surface, e.g., a table or a sidewalk. Therefore, when an object is copied from a source to a target image, our system aims to orient it in such a way that its support surface in the source becomes aligned with an appropriate support surface

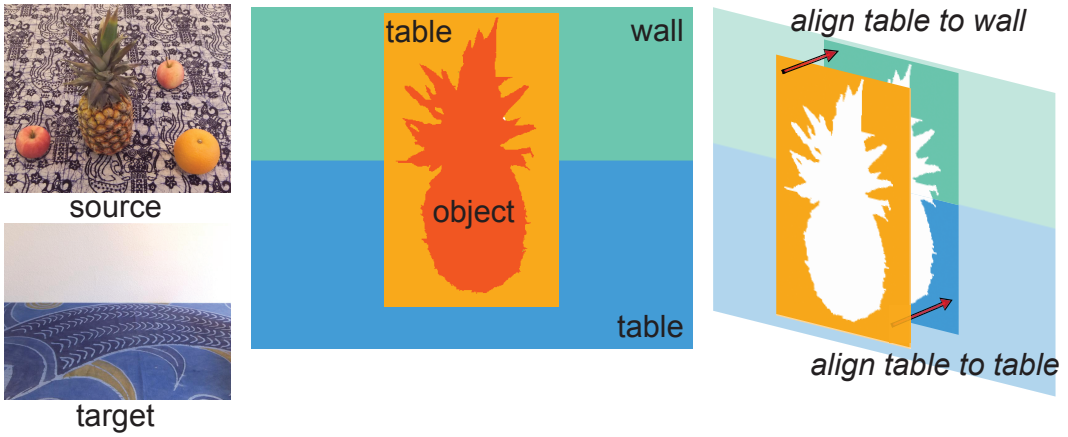


Figure 6.4: Local Surface Orientation Alignment. ***Left*** *Source and target scene images (left eye) with different orientations of the support surfaces.* ***Middle+Right*** *After the pineapple is copied and pasted into the target scene, we compute a transformation for best alignment. In this case, there are two possible alternatives, but the best choice would be to align the source's table surface to the target's table surface.*

in the target. As an example consider the situation in Figure 6.4. When the pineapple from the source scene on the left is copied into the target on the right, we aim to align the supporting table surfaces. This registration problem could be solved using a general point cloud registration technique [Besl and McKay, 1992]. We observe, however, that in practice objects are mostly placed onto planar support surfaces. Therefore we use a simple strategy to align supporting planes.

During the Selection step the images have been segmented, and each foreground object and background surface is represented by a segment. For a selected object in the source we define a set $\mathcal{S}$ of neighbor segments. For example in Figure 6.4 the pineapple has the table as its neighbor segment. When the object is pasted into the target, $\mathcal{S}$ will overlap with a set $\hat{\mathcal{S}}$ of segments in the target scene. In the example of Figure 6.4, $\hat{\mathcal{S}}$ contains the target's table and wall segments. Exploiting the fact that support surfaces typically are planar, we estimate a least squares fitting plane for each $s \in \mathcal{S}$ and $\hat{s} \in \hat{\mathcal{S}}$. For each segment in $\{\mathcal{S}, \hat{\mathcal{S}}\}$ we define a coordinate frame $(\mathbf{R}, \mathbf{t})$, with rotation $\mathbf{R} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and translation $\mathbf{t}$. We define $\mathbf{t}$ as the centroid of the 3D points associated with the segment, and $\mathbf{R}$ is computed from the normal of the estimated plane. We then aim to find the two segments s^* and $\hat{s}^*$ with the most similar orientation, i.e., they minimize the rotation required to align the

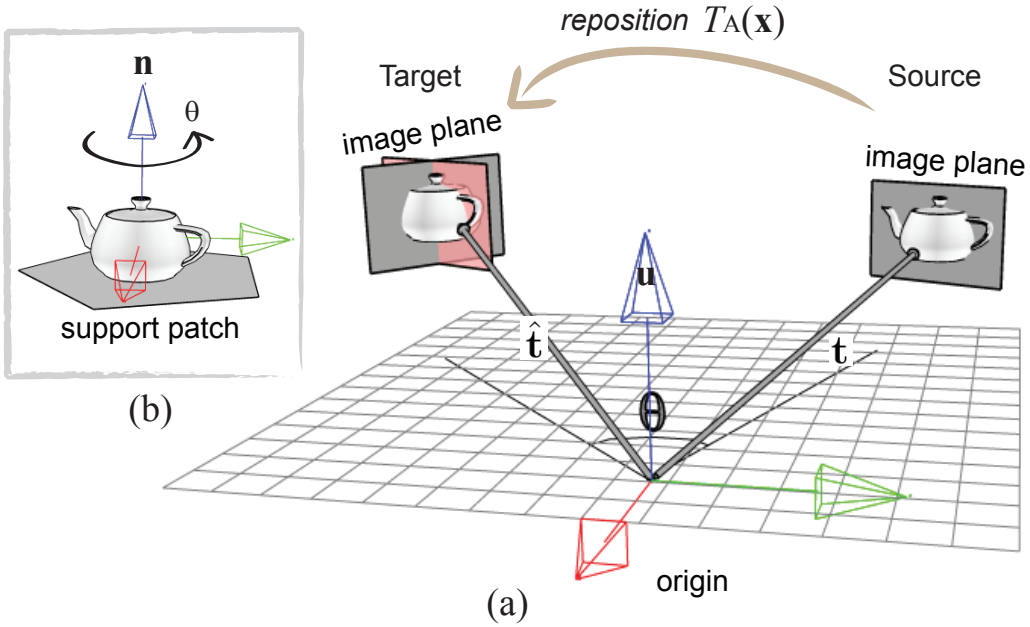


Figure 6.5: Rotation Constraints. ***(a)*** *Due to missing data (shaded in red), only part of the repositioned object can be rendered onto the composited image.* ***(b)*** *We compensate the perspective change by rotating the object back, around the normal of the support patch, so that most of the available data still faces the viewer.*

source and target segment:

$$(s^*, \hat{s}^*) = \arg \min_{s \in \mathcal{S}, \hat{s} \in \hat{\mathcal{S}}} \| \mathbf{R}_{\hat{s}} \mathbf{R}_s^{-1} \|. \tag{6.2}$$

The desired alignment transformation $\mathbf{T}_A(\mathbf{x}) = \mathbf{R}_A(\mathbf{x}) + \mathbf{t}_A$ is the transformation that aligns these two segments. It can be computed as

$$\begin{aligned} \mathbf{R}_A &= \mathbf{R}_{\hat{s}^*} \mathbf{R}_{s^*}^{-1}, \\ \mathbf{t}_A &= \mathbf{t}_{\hat{s}^*} - \mathbf{R}_A(\mathbf{t}_{s^*}). \end{aligned} \tag{6.3}$$

Instead of using the entire segments, in practice we only use information from partial segments. Partial segments are determined by taking a predefined area around the selected object, e.g., the rectangular orange area around the pineapple in Figure 6.4. We denote such partial segments as patches.

Rotation Constraints

If one could move an object freely in 3D, parts that previously were hidden would become visible, as shown in Figure 6.5a. With stereoscopic input images we have no data available for the invisible parts and hence, in-painting or object completion techniques would be required to handle such rotations. However, in-painting and object completion are difficult tasks. Instead, we constrain the rotation to keep the object's "forward facing" orientation of the source images. We accomplish this by rotating the object around the normal of the support plane computed during the alignment step.

Assume $\mathbf{t}$ is the centroid of the object in the source scene, and $\hat{\mathbf{t}}$ is its new location after being pasted into the target scene. With the alignment transformation in Equation 6.3 we get:

$$\hat{\mathbf{t}} = \mathbf{T}_A(\mathbf{t}) = \mathbf{R}_A(\mathbf{t}) + \mathbf{t}_A. \tag{6.4}$$

We denote the up vector of the camera as $\mathbf{u}$, and determine the angle θ between the projections of $\mathbf{t}$ and $\hat{\mathbf{t}}$ onto the ground plane (see Figure 6.5(a)). We can then apply a corresponding rotation $\mathbf{R}_F$ to ensure a target orientation as close as possible to the source orientation of the object. $\mathbf{R}_F$ is defined as:

$$\mathbf{R}_F = \theta\mathbf{n}, \tag{6.5}$$

where $\theta\mathbf{n}$ is the so-called Euler axis–angle representation, and $\mathbf{n}$ denotes the normal of the support segment, as shown in Figure 6.5b. We can compute θ as:

$$\theta = \sin^{-1}\left(\frac{\|(\mathbf{t} - (\mathbf{t}\cdot\mathbf{u})\mathbf{u}) \times (\hat{\mathbf{t}} - (\hat{\mathbf{t}}\cdot\mathbf{u})\mathbf{u})\|}{\|\mathbf{t} - (\mathbf{t}\cdot\mathbf{u})\mathbf{u}\|\|\hat{\mathbf{t}} - (\hat{\mathbf{t}}\cdot\mathbf{u})\mathbf{u}\|}\right), \tag{6.6}$$

In other words, we rotate the object around $\mathbf{n}$ at its centroid $\hat{\mathbf{t}}$ with angle θ. The rotation constrained result may not be fully satisfying to the user and we thus provide additional user control over the rotation for each pasted object in the scene.

Stereo Billboards

The transformations T_A and R_F from above determine the desired pose of the selected object copied into in the target. Due to the inaccuracies in the computed disparities, the objects' corresponding 3D point clouds are not suitable for direct rendering. To overcome this problem we adopt the motivation from Liu et al. [2009a] to compute parametric warps for rendering. We

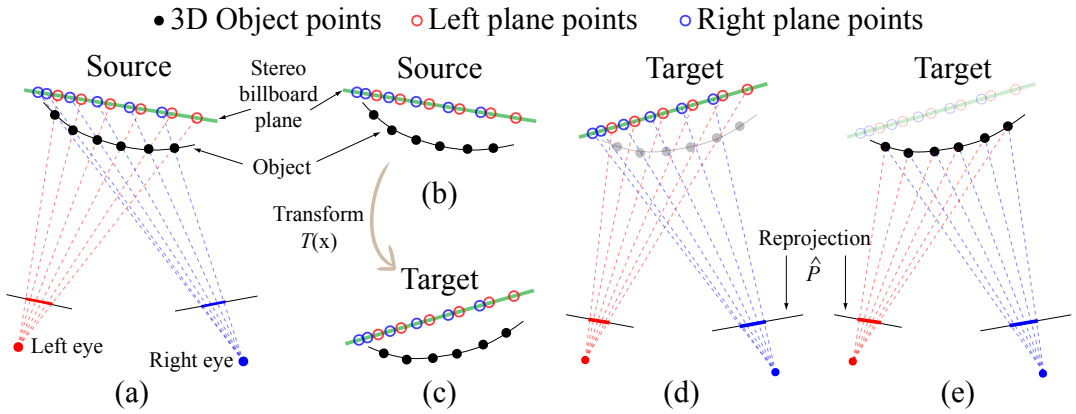


Figure 6.6: Stereo Billboards. *This figure illustrates the computation of the stereo billboard plane as the planar proxy geometry. An object is represented by a set of 3D points, with corresponding pixels in the left (red) and right (blue) eye source images.* ***(a)*** *Pixels are back-projected onto a current estimate for a stereo billboard plane (green).* ***(b, c)*** *Stereo billboard points and object points are transformed to the target scene.* ***(d, e)*** *Transformed stereo billboard and object 3D points are projected onto the target left and right eye image. The optimal stereo billboard plane minimizes the difference between projected points in (d) and (e).*

approximate the 3D point clouds with planar proxies, and we compute homographies for the left and right eye as our parametric warps for rendering. However, the stereo volume of the source object is implicitly encoded by the 3D point cloud, and representing them by a plane could make the composited object appear flat: the so-called cardboarding effect in stereo. In order to preserve the stereo volume of the source objects in stereoscopic 3D, we introduce an approach we call stereo billboards. The goal is then to determine a *single* planar proxy, such that the error between points projected by the parametric warp, and points from the projected 3D point clouds, is minimized.

We define stereo billboards as finding an optimal common plane $\mathbf{v}$, from which a pair of consistent homographies can be computed. Figure 6.6 sketches a particular configuration. We denote pixels of a segmented object in the left and right source images as $\mathbf{l}$ and $\mathbf{r}$ respectively. Each pixel pair $(\mathbf{l}_i, \mathbf{r}_i)$ has an associated 3D point $\mathbf{X}_i$. For a given plane parametrization we can project the points $\mathbf{X}$ onto the plane $\mathbf{p} = (\mathbf{v}^T, 1)$ resulting in $\tilde{\mathbf{X}}$, such that,

$$\begin{aligned} \mathbf{v}^T\tilde{\mathbf{X}}_i + 1 = 0 \qquad & \mathbf{P}^l\tilde{\mathbf{X}}_i = \mathbf{l}_i, \\ \mathbf{v}^T\tilde{\mathbf{X}}_i + 1 = 0 \qquad & \mathbf{P}^r\tilde{\mathbf{X}}_i = \mathbf{r}_i \end{aligned} \tag{6.7}$$

where $\mathbf{P}^l$ and $\mathbf{P}^r$ denote the camera projection matrices for the source image pairs.

Given Equation 6.3 and 6.6 we can define $\mathbf{T}(\mathbf{X}) = \mathbf{T}_A(\mathbf{R}_F(\mathbf{X}))$. We can then solve the following minimization problem:

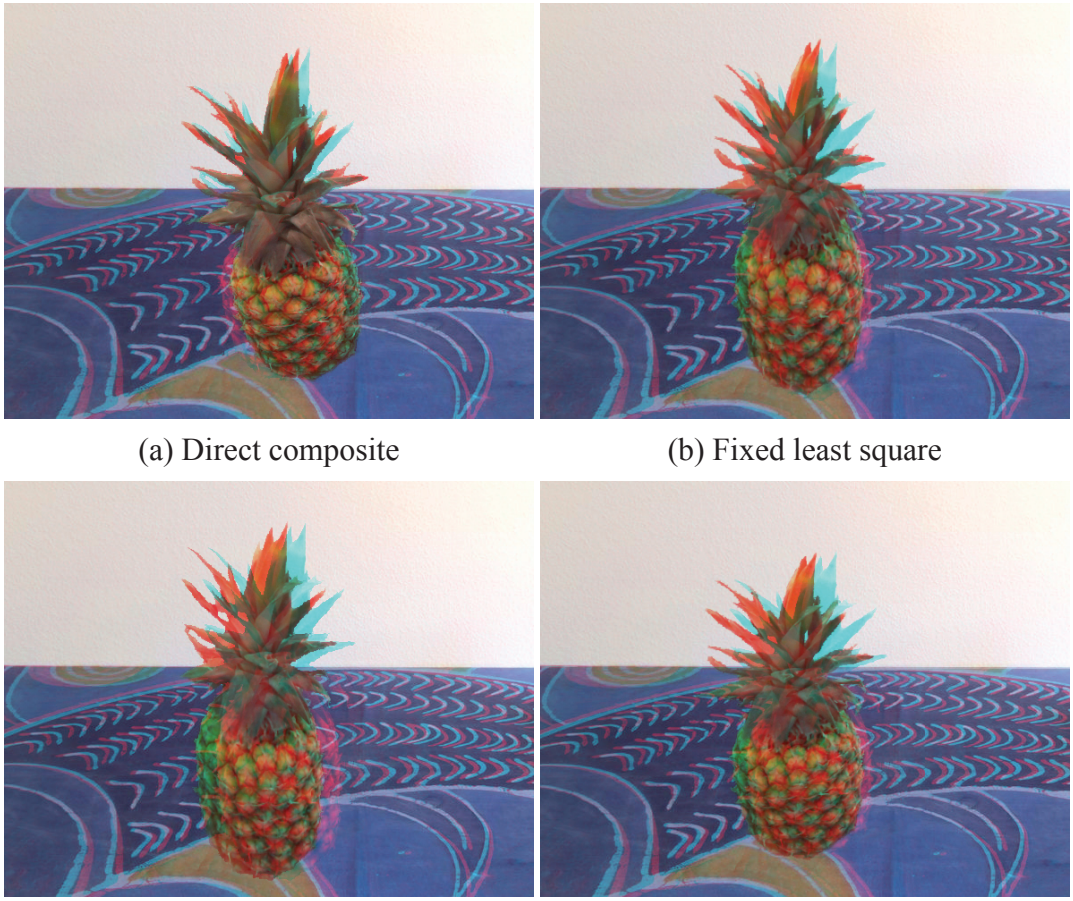
$$\mathbf{v}^* = \arg\min_{\mathbf{v}} \sum_i \Big(\|\hat{\mathbf{P}}^l \mathbf{T}(\mathbf{X}_i) - \hat{\mathbf{P}}^l \mathbf{T}(\tilde{\mathbf{X}}_i)\|^2 + \|\hat{\mathbf{P}}^r \mathbf{T}(\mathbf{X}_i) - \hat{\mathbf{P}}^r \mathbf{T}(\tilde{\mathbf{X}}_i)\|^2 \Big), \tag{6.8}$$

where $\hat{\mathbf{P}}^l$ and $\hat{\mathbf{P}}^r$ denote the camera projection matrices for the target image pairs. Equation 6.8 aims to find the optimal common plane which minimizes the image space difference between the original points and the plane approximated points, in order to faithfully represent the stereo object during rendering. Figure 6.7 compares direct compositing, straightforward least squares fitting, and our common plane optimization of Equation 6.8. Stereo billboards can better preserve the stereo volume of the object.

To solve Equation 6.8 in the presence of inaccuracies in the disparity map, we incorporate an outlier removal step by performing an erosion on the 2D image pixels and remove the corresponding 3D points. While this does not guarantee that all outliers will be removed, in practice we found that the resulting common planes that were fitted to the remaining points gave acceptable results.

Occlusion

Composition of pasted objects behind other objects requires the correct handling of occlusions. Furthermore each segmented object has an associated alpha matte for handling the mixed pixels along the segmentation boundaries. Therefore, to ensure the correct order for occlusions and transparencies in rendering, we have to perform depth sorting on the objects. Methods that require per-pixel depth values for depth sorting, e.g., depth peeling [Mammen, 1989], lead to interweaving objects due to the inaccuracies in the computed depth maps. We instead use the planar proxies of Section 6.2.3 for depth sorting. The overhead of having to recompute the proxy ordering is negligible since we typically only have a limited number of planes to consider in the ordering. We can interactively move the pasted layers while correctly handling the occlusions for the composite, see Section 6.3 for more details.



(a) Direct composite (b) Fixed least square

(c) No stereo volume (d) Dynamic fitting

Figure 6.7: Stereo Billboards Compositing. ***(a)*** *Direct composite result with source images from Figure 6.4.* ***(b)*** *Result with least square fitted plane proxy.* ***(c)*** *Same as (b), but now using only a single image for both left and right eye emphasizes the "cardboard" effect.* ***(d)*** *Our stereo billboards using dynamically optimized common plane. Our result better preserves the stereo volume.*

Shadow Synthesis

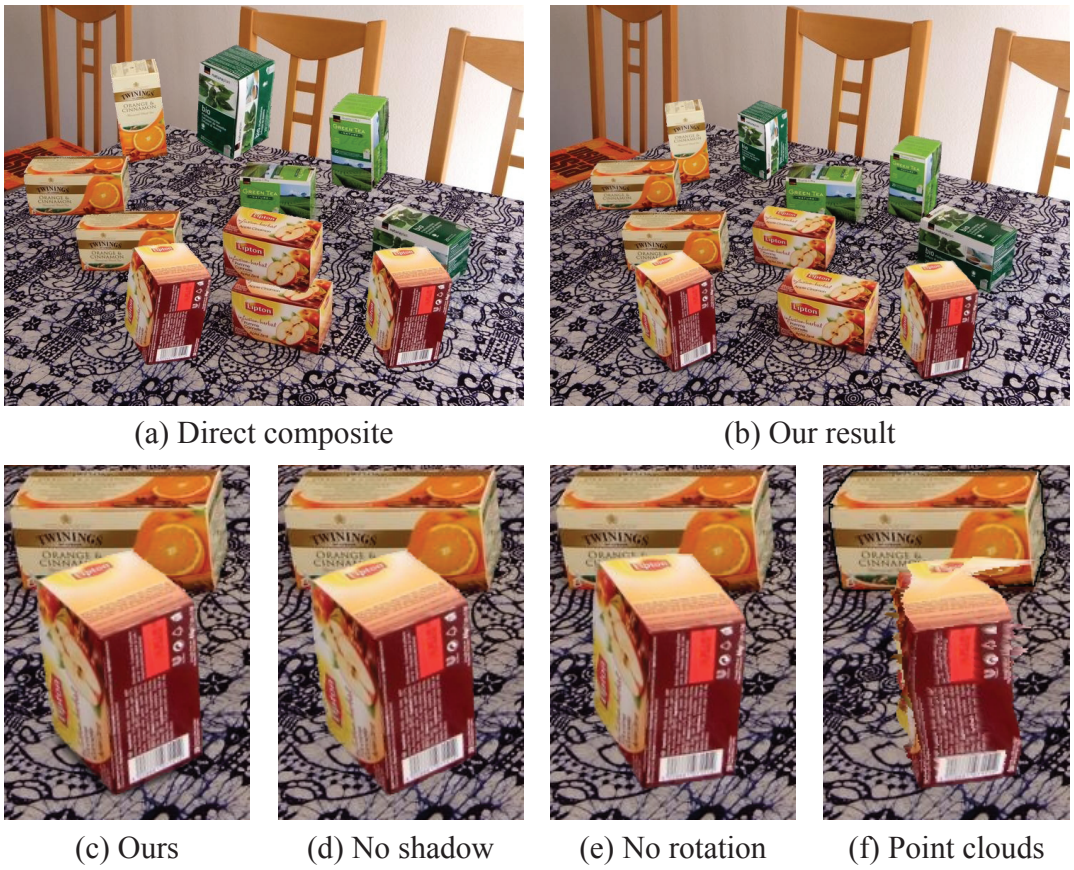
Shadows are an important cue for judging contact between surfaces. In the absence of knowledge about the light direction in the scene, we approximate contact shadows by using screen-space volumetric ambient occlusion [Loos and Sloan, 2010]. A depth map of the composite scene is required to synthesize the shadows. We could use the point clouds to obtain depth images, but this would be inconsistent with the stereo billboard warp. Therefore, we obtain disparities for the composited scene directly from the warp instead



(a) No shadow (b) With shadow (c) Depth map

Figure 6.8: Shadow Synthesis. ***(a)*** *Even objects with the same orientation, when copied & pasted into the same scene, will appear to be floating in the absence of contact shadows.* ***(b)*** *Synthesized contact shadows generated with our method.* ***(c)*** *Depth map used for shadow synthesis. Note: we only render the selected object and the underlying surfaces into the depth buffer.*

and achieve more accurate shadows. Let $(\mathbf{l}_i, \mathbf{r}_i)$ and $\mathbf{X}_i$ denote a pair of corresponding points on the selected object in the source images and their associated 3D point respectively. Using $\mathbf{v}^*$ from Equation 6.8 we can project $\mathbf{X}_i$ onto $\mathbf{v}^*$ to obtain $\tilde{\mathbf{X}}_i$, and compute $\hat{\mathbf{l}}_i = \hat{\mathbf{P}}^l \mathbf{T}(\tilde{\mathbf{X}}_i)$. The image points $(\hat{\mathbf{l}}_i, \hat{\mathbf{r}}_i)$ denote the new positions of $(\mathbf{l}_i, \mathbf{r}_i)$ in the composite target images. We then render the disparity value $\hat{\mathbf{r}}_i - \hat{\mathbf{l}}_i$ at pixel $\hat{\mathbf{l}}_i$ into the depth buffer of the left composite image, and vice versa for the right one. This method better preserves contours in the depth map and is also more consistent with the stereo volume. Synthesized shadows therefore exhibit less noise and better approximate the object. A computed depth map with our method is shown in Figure 6.8c.



(a) Direct composite (b) Our result

(c) Ours (d) No shadow (e) No rotation (f) Point clouds

Figure 6.9: Composition. ***(a)*** *The user can roughly place the objects into the target scene.* ***(b)*** *Our system will automatically arrange the objects in a right perspective and depth order.* ***(c)*** *Close-up of a region in (b).* ***(d)*** *Without shadow, the object appears to be floating.* ***(e)*** *Without the rotation constraint, the copied object edges are not parallel with target object edges, resulting in unnatural results.* ***(f)*** *Rendered results with point clouds show artifacts due to inaccurate depth reconstruction and missing data.*

6.3 Results

For all results presented in this section, we used the color transfer method described by Reinhard et al. [2001]. The objects are selected and copied from different source images shown in Figure 6.12.

Figure 6.9 compares direct composition (a) with our approach (b). Our system can handle multiple objects composed in depth, and occlusions are continuously updated while the user determines a final location of the copied ob-



Figure 6.10: Results. *Examples of composed scenes using objects copied from the source images of Figure 6.12.* ***a—c*** *Left eye images.* ***d—f*** *Anaglyph results.*

jects in the target scene. Synthesized shadows and local surface alignment are necessary to make the composition plausible (Figure 6.9c vs. Figure 6.9d,e). Direct rendering with of the 3D point cloud derived from the depth map makes the need for using stereo billboard evident.

Figure 6.10a—c demonstrate additional examples of objects copied onto vari-

ous surfaces. Local surface alignment adjusts orientation and scale of the objects. Figure 6.10d—f shows anaglyph results of our copy & paste approach to demonstate that objects are correctly composited in depth and do not appear flat. Figure 6.10f shows that although overall composition is correct with respect to orientation, scale and depth, the illumination difference between the source and target scene makes the copied object stand out.

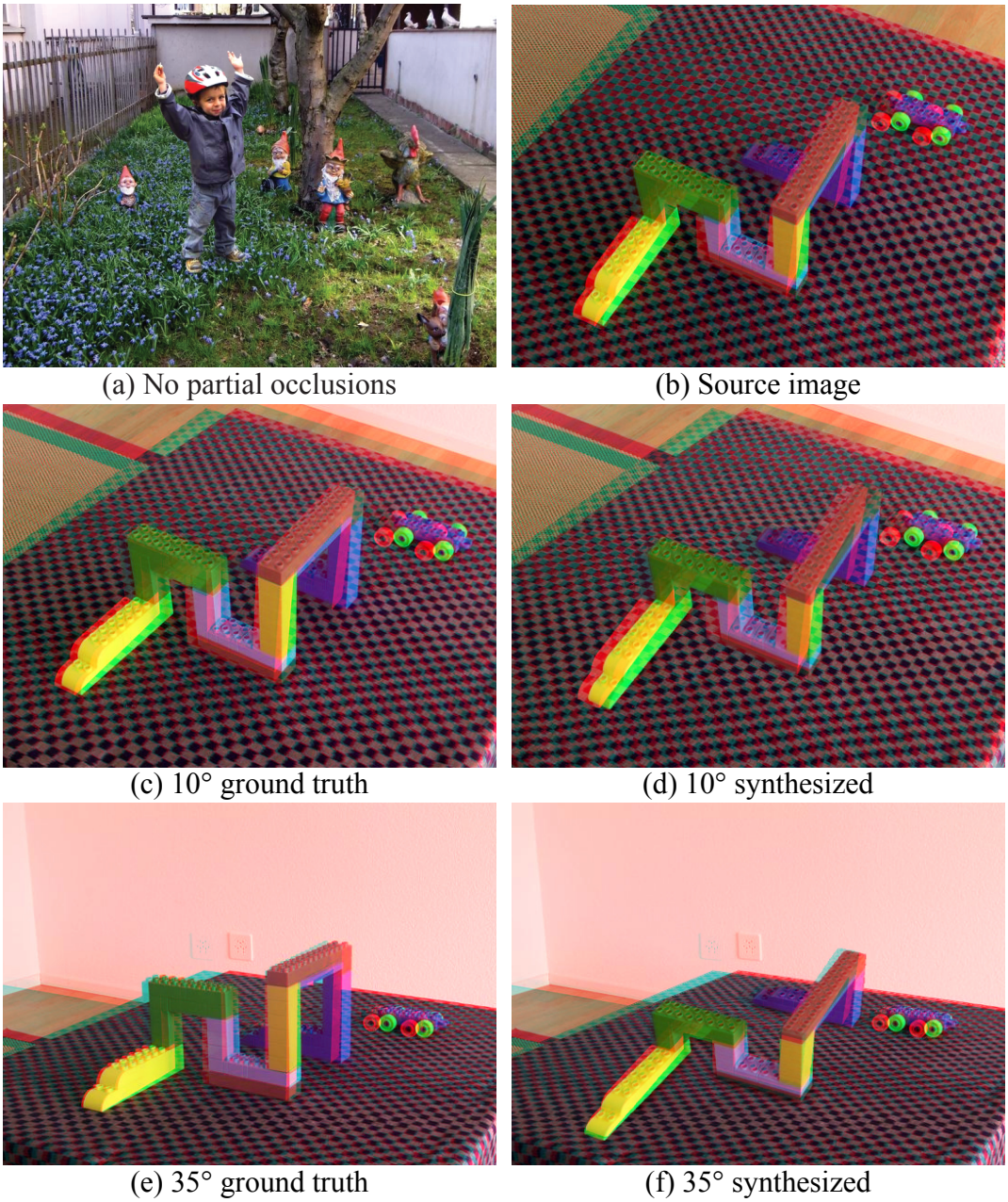
6.4 Discussion

We have presented an end-to-end system for 3D copy & paste, which extends 2D copy & paste editing for still images to stereoscopic 3D. Our proposed system and methods build on previous work for computing depth maps and performing segmentations. To address inaccuracies in the current depth maps, we have specifically aimed to make the segmentation refinement, segmentation transfer, alignment, stereo billboards, and occlusion methods robust to those inaccuracies. As explained, this is largely achieved by approximating the objects' associated 3D point clouds with proxy geometry. For simplicity, we currently use planar proxy geometry.

Approximating geometry with planar proxies has certain limitations. Planar proxies do not preserve detailed depth structure, such as the grass surface in Figure 6.11a. As a result, the lack of partial occlusions makes the copied object appear to float.

Another limitation of planar stereo billboards is that they may no longer respect epipolar geometry. This might result in vertical disparities that could strongly interfere with the stereopsis. To evaluate the amount of vertical disparity that is introduced, we use an object which is not well represented by a plane, shown in Figure 6.11b. The object is copied from the source scene into two target scenes with the support surface at a different orientation. Figure 6.11c shows the ground truth image for 10° rotation, and Figure 6.11d shows the composition using our stereo billboards. The vertical disparities in this case are around 0.8% of the object height. Figure 6.11e,f compares ground truth and our stereo billboards for 35° rotation. In this case the vertical disparities are around 2.4%. Fukuda et al. [2009] report a tolerance of 45 arcmin for random dot stereograms. For 100 dpi display viewed at a distance of 50 cm, this amounts to a vertical disparity tolerance of about 26 pixels. The vertical disparity for our 35° case is about 10 pixels, which is well within the reported tolerance, however a more thorough analysis should be conducted.

Instead of planar proxies, piece-wise planar proxies may be fitted to the 3D points and depth to better approximate the objects. Avoiding proxy geom-



(a) No partial occlusions (b) Source image

(c) 10° ground truth (d) 10° synthesized

(e) 35° ground truth (f) 35° synthesized

Figure 6.11: Limitations. ***(a)*** *The lack of fine depth detail after planar approximation makes the copied object appear to float.* ***(b)*** *Source image for comparisons of orientation changes.* ***c,d; e,f*** *Ground truth vs. our approach for 10° and 35° orientation change. For 35°, the epipolar geometry is no longer correct, but the stereo images can still be fused.*

etry altogether would require in-painting [Wang et al., 2008]. High quality

in-painting however, is a difficult task, performed off-line and therefore typically limited to only relatively small areas.

Stereo billboards help to preserve the stereo volume of the copied source object. However, if the initial depth volume in the source image is relatively flat, such as for narrow baselines (or interocular), stereo billboards will not be able to increase the stereo volume in the target. Furthermore, for large differences in baseline between source and target, stereo billboards may not be able to preserve volume. In particular achieving artistic stereo effects such as hypostereo (gigantism) and hyperstereo (miniaturization) [Koppal et al., 2010] in copy & paste is an interesting topic for future work. We may be able to exploit the work by Lang et al. [2010] in such scenarios.

For plausible appearance of copied objects, we approximate contact shadows to avoid objects from appearing to float. However, illumination differences between the source and target images is a larger problem that we did not address. This problem is not specific to 3D, see for example [Lalonde et al., 2007]. Although we use the color transfer method described by Reinhard et al. [2001], this does not always give the desired results. For truly plausible appearance of pasted objects, more information about the scene illumination should be recovered, and exploited to relight the objects. The depth map could then also be used for shadow casting and light attenuation. However, relighting is an active area of research with no good solution to date.

We have demonstrated stereoscopic 3D copy & paste for still images, which achieves high quality compelling composition results and convincing stereo viewing. An interesting direction for further exploration would be to extend our approach to stereoscopic 3D video contents. Depth reconstruction, segmentation, alignment, occlusions, and depth composition will now all have to be done for dynamic objects and scenes. We have already partially addressed some of these challenges in previous chapters in this thesis. Finally, with the rapid growth in popularity of 3D the need for stereoscopic 3D compositing tools in general will grow as well. We hope that our system serves as a start in the exploration of more general stereoscopic 3D compositing tools.



Figure 6.12: Source Images. *Input source images for copy & paste.*

CHAPTER 7

Stereoscopic 3D Display

The final step in the stereoscopic 3D pipeline is the display of stereoscopic image pairs. The images intended for the left and right eye are either shown simultaneously or temporally multiplexed at high refresh rates. Although stereoscopic 3D display systems continue to improve, some problems still remain. In this chapter we address the problem of crosstalk or *ghosting*. Since the perception of depth for stereoscopic 3D relies on the human visual system, we propose a perceptually-based method to compensate the input images and eliminate the perception of ghosting. Many computational models have been developed using psycho-physical and physiological experiments and which model various properties of the human visual system. We propose to incorporate some of these models into an optimization-based perceptual compensation. The inclusion of perceptual models results in a perceptually more optimal, smooth distribution of the ghosting in the locally surrounding areas. This smooth distribution can eliminate the perception of ghosting. Our results are evaluated with a user study which shows that our method is preferred over previous approaches.

Our perceptual compensation is aimed at compensation for ghosting, also referred to as *deghosting*. However, we will argue that our perceptual compensation can be applied to compensate for additive unintended illumination in display systems in general. We demonstrate this by applying perceptual

compensation to scattering in immersive display systems. Our focus is on high-end systems such as those found in cinema environments, but at the end of the chapter we also discuss our perceptual compensation with respect to consumer level display systems.

7.1 Motivation

Entertainment display system installations, such as cinemas, are among those with the highest quality demands. Many movie theaters either have been, or are in the process of being, converted to digital projection systems. High-end projectors are employed to ensure a large peak intensity at maximum contrast. Digital projection systems have proven to be very suitable for stereoscopic 3D motion pictures. Stereoscopic image pairs are either temporally or spatially multiplexed. An example of temporal multiplexing is the opto-electronical switching of the polarization direction between the left and right eye images [Lipton, 2012]. With corresponding polarization filters for the eyewear, the images can be appropriately filtered for the left and right eye of an observer. An example of spatial multiplexing is the separation of the visible wavelengths in the color spectrum into different narrow wavelength bands for the left and right eye images [Jorke and Fritz, 2003]. Again, corresponding filters for the eyewear can appropriately filter the left and right eye image.

The different components used in stereoscopic 3D cinemas, such as the filters and the screen, cannot perfectly separate the images for the left and right eye. For stereoscopic cinemas to be cost-effective, the eyewear needs to be low-cost, which further reduces the separation power and exacerbates the problem. As a result of non-perfect separation, the image intended for one eye is *contaminated with* or *polluted by* a small amount of light coming from the image intended for the other eye. This *light pollution* in stereoscopic 3D displays is referred to as ghosting or crosstalk. Figure 7.1 shows an example of ghosting, where a dim copy of the image intended for the other eye can be observed.

We note that ghosting is a form of additive light pollution. Another example of such light pollution is the scattering, or reflection into multiple directions, of light onto other areas of a screen in concave display surfaces. We define light pollution as light that is originally injected as intended light, but by some physical property of the display system results in unintended light contribution to (portions of) the intended image. In the general case light pollution results in a reduction of contrast, which can lead to a loss of detail in

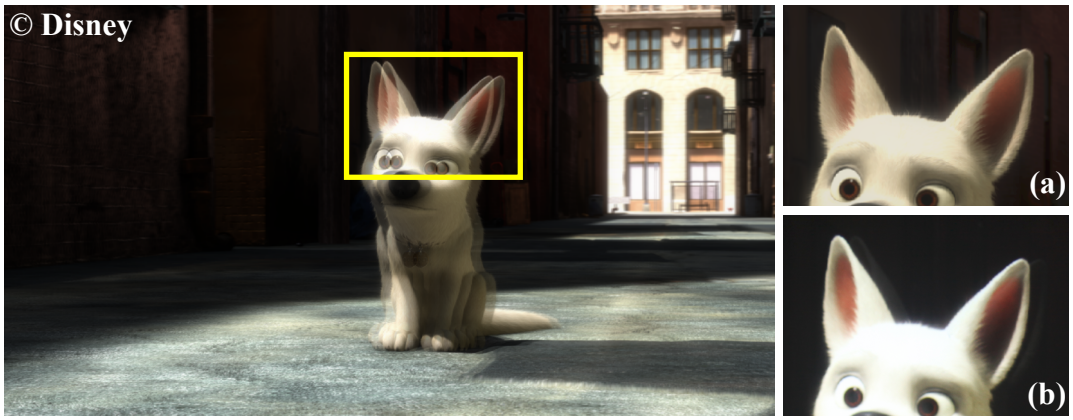


Figure 7.1: Ghosting. *An example of ghosting:* ***Left*** *Superimposed left and right eye images reveal the shift due to the disparities.* ***Right*** *(a) The original input image for the right eye. (b) The image observed after projection for the right eye acquired with a camera through an eyewear polarizing filter. A dim copy of the left image can be observed. (Best viewed electronically, adjusting brightness and gamma settings if images appear too dark.)*

the images. However for ghosting it could lead to the more severe problem of making the stereoscopic 3D viewing experience uncomfortable. Ghosting may interfere stereopsis when unintended (depth) edges conflict with intended (depth) edges, thereby hindering the proper fusion of stereo images [Kooi and Toet, 2004, Tsirlin et al., 2011]. Combined with the vergence-accomodation conflict [Hoffman et al., 2008], it is imperative for stereoscopic displays that ghosting is minimized.

Our discussion is focused on stereoscopic 3D cinema displays. However, besides cinema many consumer display devices, even mobile ones, are now also capable of showing stereoscopic content. Especially consumer display devices based on polarizing filters exhibit ghosting. The need for ensuring good quality stereoscopic content and a comfortable viewing experience therefore extends beyond the cinemas.

In this chapter we present a perceptually-based compensation method which takes properties of the human visual system into account. Our focus is on the compensation of ghosting, but by applying perceptual compensation to the problem of scattering, we demonstrate that our method can be applied to additive light pollution in display systems in general. We compare our results with other approaches and show that with our method image details are better preserved and contrast is reduced only locally in the images. Before we

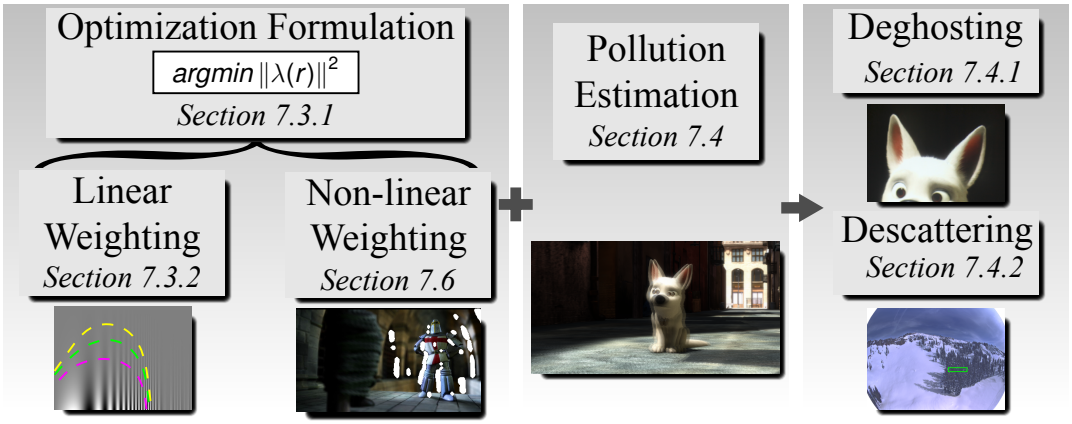


Figure 7.2: Overview. *Overview of perceptually-based compensation. The formulation is generally applicable for compensation of light pollution, and we discuss two applications: deghosting and descattering.*

discuss the details of our method, we first discuss the occurrence of ghosting in various types of display systems.

7.2 Ghosting

Stereoscopic displays are often categorized as active or passive. Active displays rely on the synchronization of the eyewear with the display. The separation of the left and right eye images occurs by actively obscuring via shutters the opposite eye while the current eye image is being displayed. This procedure is repeated in alternating succession. Passive stereoscopic displays on the other hand do not rely on any synchronization between the display and the eyewear. The left and right eye images are separated using matching filters, e.g. polarization filters, for both the display and eyewear. The two main advantages of passive stereoscopic displays over active ones are the reduced cost for the eyewear, and the simultaneous viewing by virtually an unlimited amount of observers.

Although active systems physically obscure the appropriate eye, ghosting could still occur due to a non-perfect synchronization between eyewear and display. Since, for the reasons mentioned above, cinemas rely on passive stereoscopic display systems our work and results are discussed related to such passive systems. However, our method could be applied to compensate for ghosting in active systems as well.

7.3 Perceptually-based Compensation

We first discuss our perceptual compensation for light pollution in general, and discuss its application for deghosting and descattering in Section 7.4. We regard light pollution as an *error* term compared to the intended or desired image. Compensation relies on the subtraction of the expected amount of pollution prior to display. Subtraction may lead to negative values in areas of low intensity in the image to be compensated. Negative values are clamped to zero and residual pollution remains. We address this by formulating the compensation as a constrained optimization problem, where the residual is weighted perceptually by incorporating models for properties of the human visual system. We discuss which models are considered and how the resulting optimization problem can be simplified to make the optimization computationally tractable. Figure 7.2 illustrates an overview of our perceptually-based compensation.

7.3.1 Compensation Formulated as Optimization Problem

Given an input image $\mathbf{x}$ to be displayed, we denote the image observed in the absence of any light pollution as the desired image $\mathbf{x_d}$. When taking light pollution into account, we denote this as the observed image $\mathbf{x_o}$. The goal of compensation for light pollution is therefore to adjust the input image $\mathbf{x}$ such that the observed image $\mathbf{x_o}$ is as close as possible to the desired image $\mathbf{x_d}$. In its simplest form compensation subtracts the amount of light pollution that is expected given the input image. Mathematically this can be expressed as a constrained optimization problem:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x_o} - \mathbf{x_d}\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1. \tag{7.1}$$

We will discuss the need for the constraints in more detail below. We introduce an observation function $\psi(\mathbf{x})$ to represent $\mathbf{x_o}$, and define:

$$\psi(\mathbf{x}) = \mathbf{x} + \varphi(\mathbf{x}). \tag{7.2}$$

Here $\varphi(\mathbf{x})$ is a function which represents the additive light pollution for the display system. The observation is thus defined as the sum of the input image and the light pollution. Combining Equations 7.1 and 7.2 we get:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\psi(\mathbf{x}) - \mathbf{x_d}\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1. \tag{7.3}$$

Since we are dealing with a physical display system, the range of intensities that can be displayed are determined by that physical system. The constraints

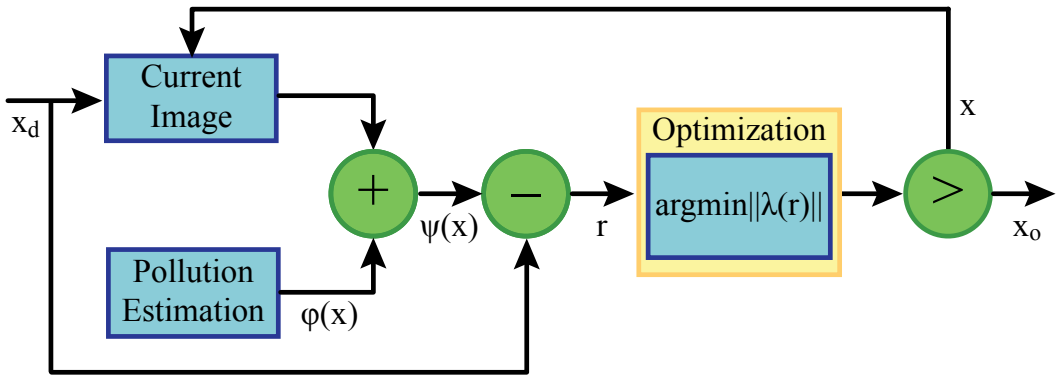


Figure 7.3: Perceptually-based Compensation Dataflow. *Schematic overview of the dataflow for our perceptual compensation.*

in Equation 7.1 are important to ensure that the pixel values remain within the range of physically attainable values. The constraint that $\mathbf{x} \geq 0$ represents the fact that we cannot have negative light, whereas the constraint $\mathbf{x} \leq 1$ represents that we cannot display more light than the maximum possible intensity (normalized to 1).

We define the difference between the observed compensated image and the desired image as the residual $\mathbf{r}$:

$$\mathbf{r} = \mathbf{x_o} - \mathbf{x_d} = \psi(\mathbf{x}) - \mathbf{x_d} = \mathbf{x} + \varphi(\mathbf{x}) - \mathbf{x_d}. \tag{7.4}$$

In the case when the pollution can be fully compensated for we have $\mathbf{r} = 0$. However in general this will not be the case, and a solution will be one that is projected onto constraints of Equation 7.3. Our goal is to compute a compensated input image $\mathbf{x}$ resulting in an observed image which is *perceptually* as close as possible to the desired image $\mathbf{x_d}$. To achieve this we should weight the residual $\mathbf{r}$ by some abstract perceptually based weighting function $\lambda()$, which can include any combination of the psychophysical and physiological aspects of the human visual system. We then write Equation 7.3 as:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\lambda(\mathbf{r})\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1. \tag{7.5}$$

A schematic overview is shown in Figure 7.3. The above formulation applies to a single channel. For color images we solve Equation 7.5 separately for each channel. This requires the images to be transformed to a color space with independent channels. In our case we use the YC_bC_r color space, which is a good approximation of the perceptually uniform *CIE-Lab* color space [Sheng et al., 2010].

It would in general be impractical, if not impossible, to solve Equation 7.5 as the perceptual weighting $\lambda()$ may be highly non-linear. To make Equation 7.5 amenable to efficient computation, we aim to linearize Equation 7.5. Unfortunately, some of the human visual system properties are highly non-linear and expensive to compute. We thus propose to model $\lambda()$ as the product of a linear and a non-linear term:

$$\lambda(\mathbf{r}) \approx \mathbf{\Lambda_n \Lambda_l r} \tag{7.6}$$

Here $\mathbf{\Lambda_n}$ represents a diagonal matrix for the non-linear term, and $\mathbf{\Lambda_l}$ a matrix for the linear term. For efficiency we compute $\mathbf{\Lambda_n}$ only once. With this approximation to $\lambda()$ we finally have:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{\Lambda_n \Lambda_l r}\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1. \tag{7.7}$$

For now we assume that $\mathbf{\Lambda_n} = \mathbf{I}$, with $\mathbf{I}$ being the identity matrix. We will defer the discussion of $\mathbf{\Lambda_n}$ until Section 7.6. We next describe how we exploit the structure in $\mathbf{\Lambda_l}$ to make the optimization computationally tractable.

7.3.2 Linear Perceptual Weighting

The question is which property of the human visual system we can exploit for the perceptual weighting. The Contrast Sensitivity Function (CSF) describes the human visual system sensitivity over the spatial frequencies as the percentage of contrast change necessary to detect a difference. On the left-hand side of Figure 7.4 are three CSFs representing photopic (yellow), mesopic (green) and scotopic (purple) viewing conditions. The sensitivity peaks around 10 cycles/degree and falls off towards the low and high frequencies. We exploit the CSF with the goal to distribute the residual $\mathbf{r}$ into areas of reduced sensitivity. , and incorporate the CSF into the optimization.

The CSF is most naturally expressed in the frequency domain, but we can represent it in matrix form as:

$$\mathbf{\Lambda}_l = \mathcal{F}^{-\mathbf{1}} \Omega \mathcal{F}, \tag{7.8}$$

where Ω is a diagonal matrix of the CSF spectral coefficients and $\mathcal{F} = \mathbf{F_y} \otimes \mathbf{F_x}$ is the two dimensional discrete Fourier transform. We can thus directly apply the CSF as the weighting function, with $\lambda(\mathbf{r}) \equiv \mathbf{\Lambda_l} \cdot \mathbf{r}$ we have:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{\Lambda_l} \cdot \mathbf{r}\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1. \tag{7.9}$$

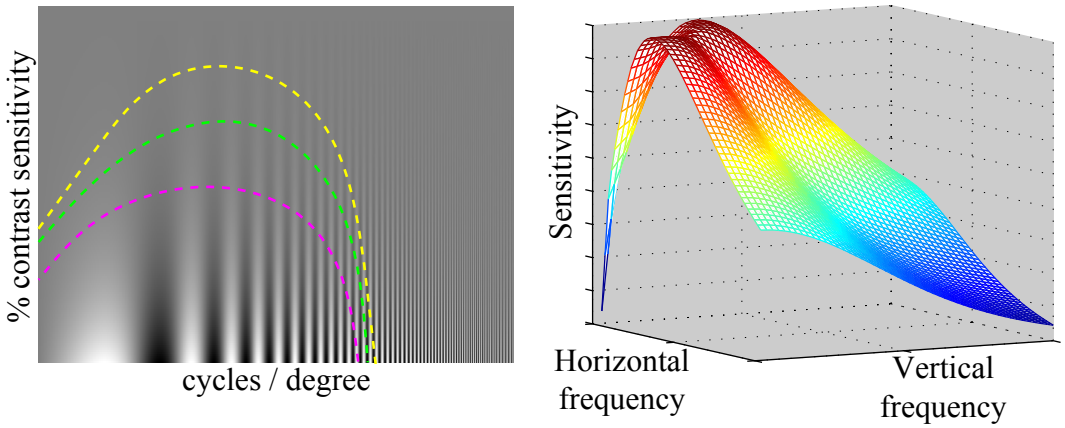


Figure 7.4: 1D and 2D Constrast Sensitivity Function. ***Left*** *CSF for photopic conditions (yellow), mesopic conditions (green) and scotopic conditions (purple).* ***Right*** *2D CSF according to the model proposed by Daly [1992].*

The challenge is that $\mathbf{\Lambda_l}$ is a dense $n \times n$ matrix, with n the number of pixels in the input image. For typical digital cinema quality content, this would result in a matrix with roughly 10^{13} non-zero elements, and would require over a terabyte of memory just to store.

By the convolution theorem, the component-wise multiplication of CSF coefficients in the spectral domain is equivalent to convolution in the spatial domain. We thus express Equation 7.9 as a convolution:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{K_l} * \mathbf{r}(\mathbf{x})\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1, \tag{7.10}$$

where $\mathbf{K_l}$ is the spatial convolution kernel corresponding to $\mathbf{\Lambda_l}$. We note that Equation 7.10 is in the class of inverse problems, e.g. non-blind deconvolution [Banham and Katsaggelos, 1997], which aim to reconstruct a signal from (possibly corrupted) observations given a known kernel. Equation 7.10 yields a linear system with a (2-level) block Toeplitz with Toeplitz blocks (BTTB) structure [Vogel, 2002]. We can exploit this fact to omit the need to store $\mathbf{\Lambda_l}$ explicitly and compute $\mathbf{\Lambda_l r}$ on demand instead.

The spatial convolution kernel $\mathbf{K}$ (omitting subscript l) from Equation 7.10 can be represented as $\text{bttb}(\mathbf{k})$, where $\text{bttb}()$ generates a BTTB matrix from $\mathbf{k}$. By taking the so-called circulant extension of $\mathbf{k}$, the $\mathbf{\Lambda_l r}$ matrix-vector products can be computed via 2D Fast Fourier Transforms (FFT) [Vogel, 2002, Ch. 5]. Given that the optimization problem in Equation 7.10 is convex, we can solve it using for example Conjugate Gradients. To satisfy the constraints we

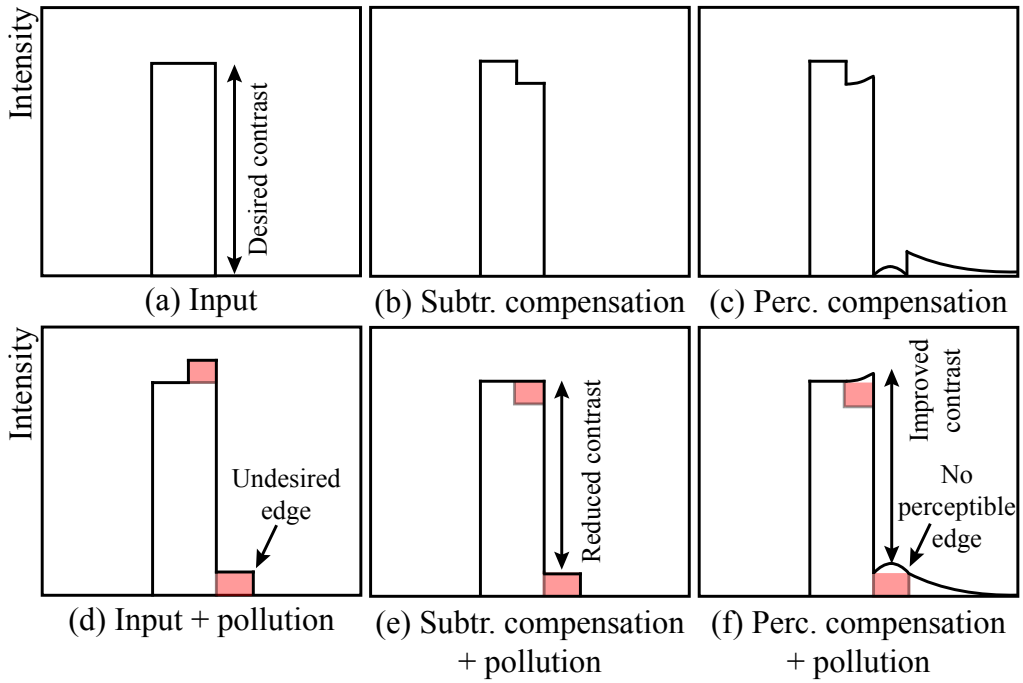


Figure 7.5: Perceptually-based Compensation. *A 1-D illustration comparing no compensation, subtractive compensation and CSF weighting (perceptual) compensation. (a)-(c) The input signals. (d)-(f) The "observed" signals. Pollution is depicted as the red shaded areas. Perceptual compensation smoothes the high contrast edge of the ghosting, while simultaneously maximizing the contrast of the original input right-hand edge.*

use the conjugate gradients with gradient projection method (GPCG) [Moré and Toraldo, 1991].

Perceptual Weighting with CSF

We illustrate the effect of weighting with the CSF in the optimization according to a 1-D example in Figure 7.5. The desired input signal is shown in Figure 7.5(a), and the input with pollution in Figure 7.5(d). The straightforward subtractive compensation reduces the input signal with the expected amount of pollution—Figure 7.5(b). However, in areas where the input signal is zero, no subtraction can be performed, and residual pollution remains—Figure 7.5(e). The weighting by the CSF distributes the residual pollution smoothly in the area local to where the pollution occurs. In addition,

the intensity of the input signal is increased smoothly to maximize the contrast—Figure 7.5(c) and (f).

7.4 Perceptually-based Deghosting and Descattering

We will next discuss two applications for our perceptually-based compensation: deghosting and descattering. For deghosting we discuss the fact that we have a stereoscopic pair of images as input. For descattering we first define scattering pollution, and then discuss how scattering pollution can be determined.

7.4.1 Deghosting

For stereoscopic display the input is a stereoscopis pair images: one the left eye ($\mathbf{x}_L$) and one for the right eye ($\mathbf{x}_R$). Therefore, for deghosting the pollution function $\varphi()$ from Equation 7.2 is a function of two input images. The observed images in the presence of ghosting are then given by:

$$\begin{aligned} \psi(\mathbf{x}_L) &= \mathbf{x}_L + \varphi(\mathbf{x}_L, \mathbf{x}_R), \\ \psi(\mathbf{x}_R) &= \mathbf{x}_R + \varphi(\mathbf{x}_R, \mathbf{x}_L). \end{aligned} \tag{7.11}$$

It is important that the pollution function is an accurate representation of the actual pollution. In the case of ghosting we can estimate the function by a series of dense measurements. We will discuss this in more detail in Section 7.5.

For solving the optimization problem using GPCG, we stack the images $\mathbf{x}_L$ and $\mathbf{x}_R$ into a single vector $\mathbf{x}$. We furthermore assume that during a single iteration of GPCG, $\varphi(\mathbf{x})$ remains constant. By re-arranging the terms in Equation 7.4 we can obtain $\mathbf{r} = \mathbf{x} - (\mathbf{x_d} - \varphi(\mathbf{x}))$. Plugging this into Equation 7.9 we get:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{\Lambda_l x} - \mathbf{\Lambda_l}(\mathbf{x_d} - \varphi(\mathbf{x}))\|^2, \text{ s.t. } 0 \leq \mathbf{x} \leq 1. \tag{7.12}$$

At the start of an iteration we first compute $\mathbf{\Lambda_l}(\mathbf{x_d} - \varphi(\mathbf{x})) = \mathbf{x}'$, then we perform one iteration of GPCG and use the current solution $\mathbf{x}$ to update $\varphi(\mathbf{x})$ and recompute $\mathbf{x}'$. We continue until convergence, or a maximum number of iterations. To compute $\mathbf{\Lambda_l x}$, we first unstack $\mathbf{x}$ and then restack the result.

7.4.2 Descattering

We also apply our perceptual compensation from Section 7.3 to the problem of descattering for immersive displays. Immersive displays employ concave screens, and light directed at a certain location reflects or scatters in multiple directions. Scattered light itself produces more scattering, referred to as multiple bounces, and produces an amount of indirect illumination which reduces contrast and degrades the intended image. One challenge with descattering is that simply evaluating the pollution term $\varphi()$ is a complex and computationally expensive process which, in the general case, involves solving the rendering equation [Kajiya, 1986]. We are specifically interested in projection-based IMAX Dome cinemas. The IMAX Dome screen are perfectly spherical and low-gain, resulting in Lambertian reflectance [Lantz, 1995, Scott, 2008]. This allows us to obtain closed-form solutions for the pollution term and an initial guess that we use for descattering in our perceptual optimization approach.

Efficient Closed-Form Pollution Estimation and Initial Guess

To compute the pollution term $\varphi()$, we exploit the fact that the point-to-point form factor within a sphere is a constant. The consequence of this is that the indirect illumination within a Lambertian sphere is spatially uniform, regardless of the projected illumination. This fact was previously used to obtain a closed-form solution to the one-bounce light transport operator within a closed, perfectly Lambertian sphere [Hawkins et al., 2005, Szirmay-Kalos, 2000]. This can be generalized to partial spherical sections and multiple bounces, which gives the following analytic expression for the pollution term:

$$\varphi(\mathbf{x}) = \frac{\rho}{4\pi r^2 - \Omega_{\mathbf{x}}\rho}(\mathbf{a} \cdot \mathbf{x}), \tag{7.13}$$

where $\rho \in (0, 1)$ is the screen gain (diffuse albedo), $\mathbf{a}$ is a vector specifying the projected area of each pixel onto the screen, $\Omega_{\mathbf{x}}$ is the projected area of the entire image, and r is the radius of the sphere. For simplicity of notation, we omit the projector-to-screen form factors, but incorporate these in our implementation. Equation 7.13 computes all bounces of indirect illumination, for all pixels, using a single dot product in $O(n)$ time where n is the number of image pixels. Hence, this computation can be performed efficiently within the inner-loop of perceptual compensation, without down-sampling, even for high-resolution input images typical of IMAX Dome projection.

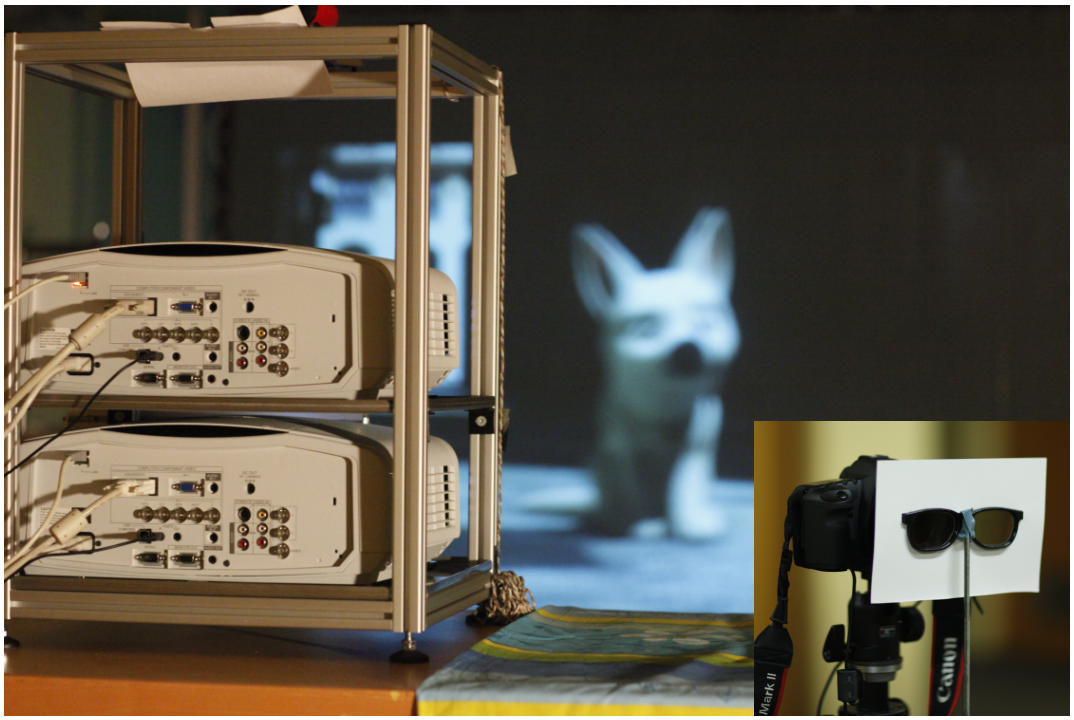


Figure 7.6: Experimental Setup. *Our experimental rear-projection setup consisting of two projectors using polarization filtering for generating separate left and right eye images. Results are captured by mounting the eyewear in front of a camera lens (inset).*

A good initial guess can improve the performance of our perceptual optimization. It can be shown that using the cancelation operator [Seitz et al., 2005] in combination with constant point-to-point form factors, subtractive compensation can be computed using a single dot product:

$$\mathbf{x} = \mathbf{x_d} - \frac{\rho}{4\pi r^2}(\mathbf{a} \cdot \mathbf{x_d}). \tag{7.14}$$

In practice we compute Equation 7.14, and if non-negative values exist, they are clamped to zero (black). With the initial guess, the optimization needs fewer iterations to arrive at a perceptual compensation.

7.5 Results

In this section we discuss the results of applying perceptually-based compensation to deghosting and descattering. We built an experimental stereoscopic display system using using two superimposed projectors and polar-

izing filters (see Figure 7.6). Projector P_L is responsible for displaying image $\mathbf{x}_L$ for the left eye, and P_R for $\mathbf{x}_R$ for the right eye. The projectors are aligned manually [1]. To correct for color and brightness differences between the projectors, we measured their *CIE-XYZ* responses using a spectraphotometer. The projectors have a gamma value of 2.2 and exhibit channel constancy, i.e. $R + G + B = W$. Color and brightness are then corrected using the common gamut mapping method [Stone et al., 1988].

Given the channel constancy, the ghosting pollution function $\varphi(\cdot, \cdot)$ is determined by a dense set of measurements for each color channel independently. First the eyewear's polarizing filter is mounted in front of the spectraphotometer, and then for a discrete set of values for each *R, G, B* the corresponding *XYZ* value is measured. We determine both $\varphi(\mathbf{x}_L, \mathbf{x}_R)$, and $\varphi(\mathbf{x}_R, \mathbf{x}_L)$. In addition to the unintended pollution we also measure the values for the intended images for the left and right eye. Compensation computations are performed in the *YCbCr* color space to avoid the correlation between the color channels in the *RGB* color space.

All our results have been generated for a "sweet-spot" location. We use the model proposed by Daly [1992] for generating a 2D CSF. As CSF parameters we use the projection resolution and size, a viewing distance of 3.0 m, light adaptation of 5 cd/m^2, and eccentricity zero.

Figure 7.7 compares no compensation (left), subtractive compensation (middle) and perceptual compensation (right) for the zoomed-in region of Figure 7.1. The input images are shown along the top row. The bottom row shows the observed images acquired by camera with our experimental setup. Subtractive compensation cannot entirely compensate for the ghosting pollution, and the ghosting is nearly as strong as for the non-compensated image. The perceptual compensation distributes the residual smoothly to make the ghosting edge imperceptible. Figure 7.8 shows additional comparisons. The top row shows the non-compensated image, while the bottom row shows zoomed-in regions for the observed subtractive and perceptually compensated images. For the perceptual compensation no visible ghost edges are observed. The ghosting for the grasshopper may appear subtle, however ghosting for the antenna makes it difficult to obtain a correct depth sensation.

Figure 7.9 shows two examples of simulated projection onto a Lambertian dome. We compare three cases: the ideal case when no scattering is present (left), residual scattering after subtractive compensation (middle), and residual scattering after perceptual compensation (right). The first and third row

[1] alignment is pixel-accurate except for the periphery of the display

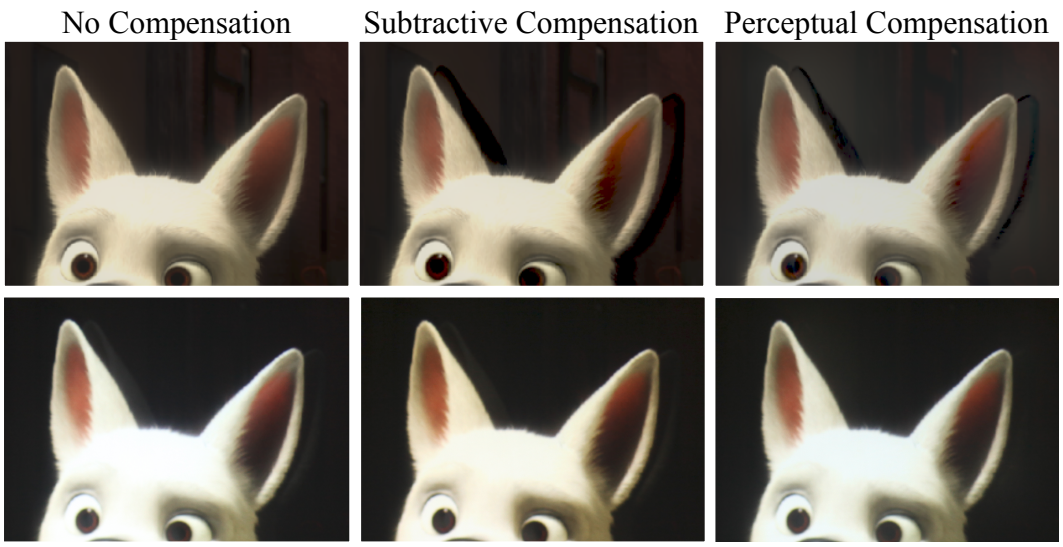


Figure 7.7: Deghosting Comparison. *Comparing the compensation for no compensation (left), subtractive compensation (middle) and perceptual compensation (right).* ***Top Row*** *Compensated input images.* ***Bottom Row*** *Observed compensated images with ghosting. Compared to no compensation, subtractive compensation can compensate for ghosting only in some areas, but ghosting edges are still clearly visible. Perceptual compensation distributes the residual such that no ghosting edges can be observed.*

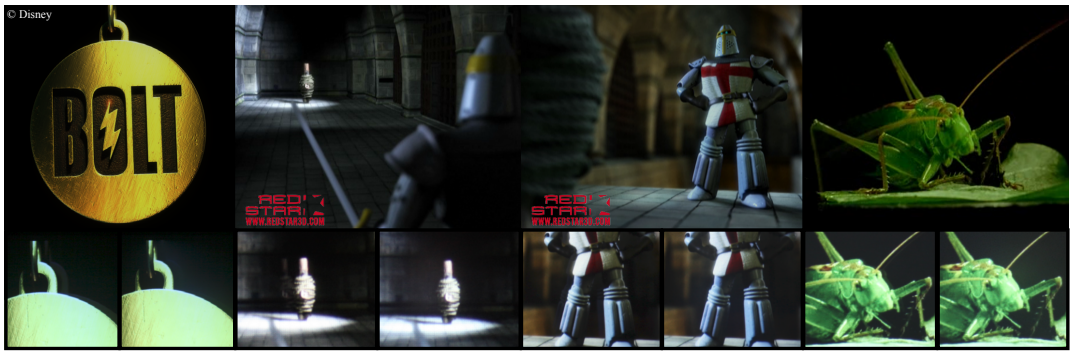


Figure 7.8: Additional Deghosting Comparisons. *Comparisons between subtractive and perceptual deghosting for the badge, indoor, knight and grasshopper scenes.* ***Top Row*** *Non-compensated input images.* ***Bottom Row*** *A side-by-side comparison of observed images acquired with a camera for subtractive compensation and perceptual compensation, for a selected area per image.*

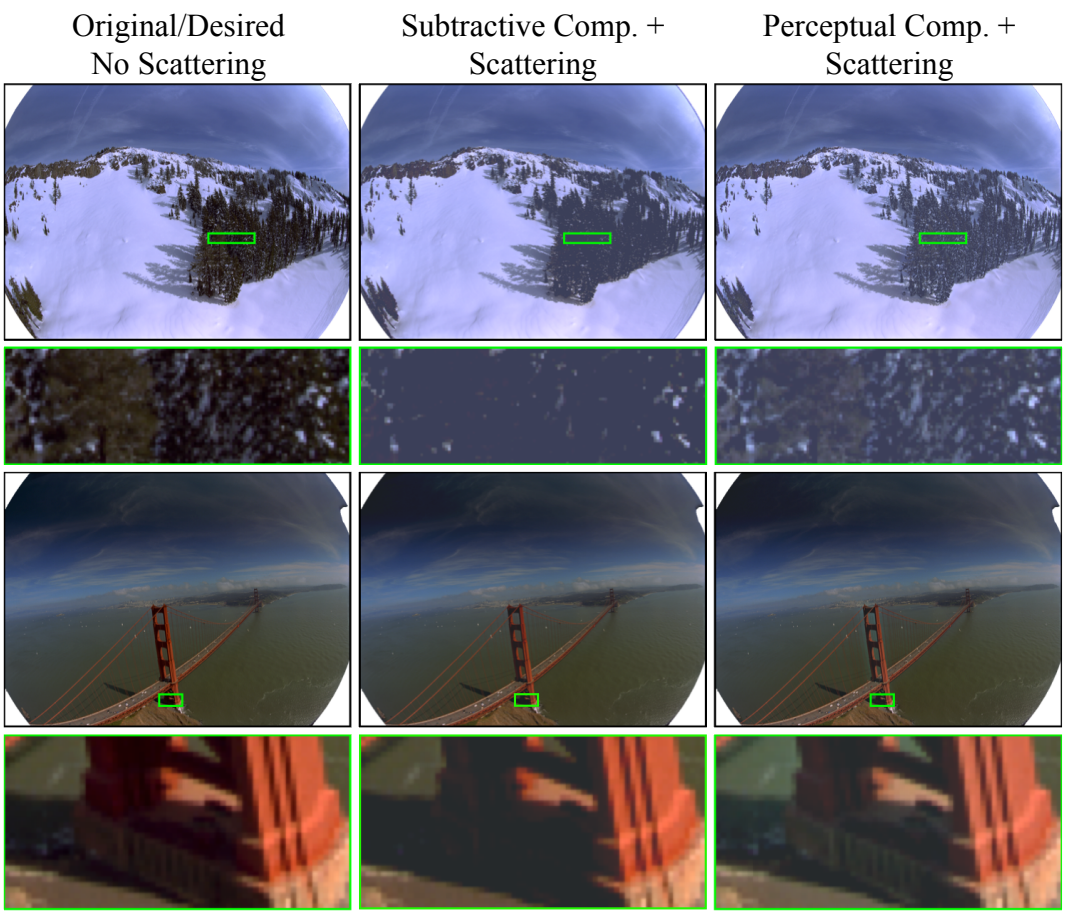


Figure 7.9: Descattering. *Simulated projection onto a spherical dome: compared to the desired image (**left**) indirect scattering reduces contrast. For areas with low intensity, subtractive compensation (**middle**) leads to negative values which are clamped to black, leading to loss of detail in the observed image. Our perceptual compensation (**right**) retains more of these dark area details with the observed image being perceptually closer to the desired image.*

show the full projected image, whereas the second and fourth row show zoomed-in areas corresponding to the rectangular demarcations in the full images. The incoming light on a spherical dome scatters in all directions, reducing the overall contrast of the projected images compared to no scattering. Subtractive compensation cannot compensate for the scattering in some darker areas, and negative values are clamped to zero (black). The zoomed-in areas in the middle column show that residual scattering reduces the contrast. Although the perceptual compensation compared to the desired image

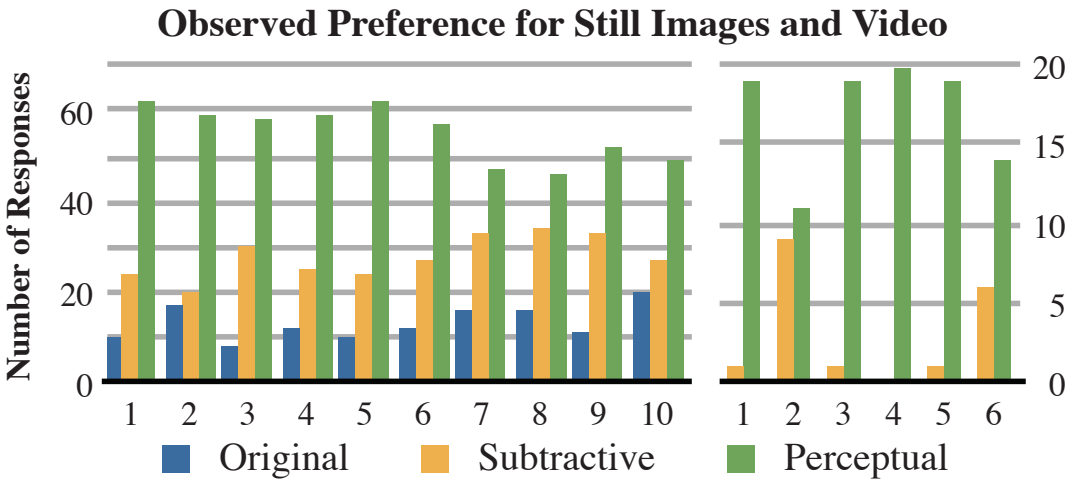


Figure 7.10: User Evaluation. *Significant preference for perceptual compensation is present in all ten images (left) and five of six videos (right) of our two separate user evaluations. For video #2 there was no statistical significant preference, which is attributed to the fact that ghosting for this case occurs in a relatively small and visually less important region of the image.*

shows some loss of detail, compared to subtractive compensation many details are preserved.

7.5.1 User Evaluation

We conducted a user evaluation to determine whether the perceptual compensation for ghosting for stereoscopic images and video is makes the viewing experience more comfortable. The experiments apply a single factor, the compensation strategy, with up to three different conditions: no compensation (original image), subtractive compensation, and perceptual compensation. We did not include compensation by raising the black level globally, as this significantly changes the image compared to the intended image. Using a forced-choice, pairwise comparison design, participants were presented balanced trials consisting of two images that differ only in compensation strategy. Participants then chose which one provides a more comfortable viewing experience.

Figure 7.10 shows that for both still images and video, there is strong statistical evidence that the perceptually-based compensation is preferred. For 10 different still images we collected from 960 total balanced trials across 16 par-

ticipants, and for 6 different videos we collected from 120 total balanced trials across 10 participants. We applied Pearson's chi-squared goodness-of-fit test to analyze the participant preferences [Sheskin, 2007]. Perceptual compensation is significantly preferred over the original image and subtractive compensation for both still images and video ($p = 0.01$ and $\chi^2(2, 960) = 283.0$, $p = 0.01$ and $\chi^2(1, 120) = 58.8$ resp.). Video #2 was the only non-significant result. Ghosting for this case occurs in a relatively small and visually less important region of the image, and is therefore less noticeable. As the ghosting does not pose strong conflicts for stereopsis, there is no statistically significant preference for subtractive or perceptual compensation.

7.6 Non-Linear Perceptual Weighting

Recall Equation 7.7. So far we have assumed that $\mathbf{\Lambda_n}$ was equal to the identity matrix. Here we extend $\mathbf{\Lambda_n}$ to incorporate additional properties of the human visual system. We observe that although pollution is physically always present, the actual pollution may be near or below the perceptual threshold of visibility. $\mathbf{\Lambda_n}$ could thus represent weights to indicate the amount that pollution is visible. Since $\mathbf{\Lambda_n}$ is a diagonal matrix Equation ref effectively turns into a weighted optimization. However, we propose to exploit $\mathbf{\Lambda_n}$ in a slightly different way.

Although we showed that we are able to avoid explicitly computing $\mathbf{\Lambda_l r}$, solving Equation 7.7 remains nevertheless computationally intensive. One possibility for increasing the performance would be to reduce the size of the problem. This could be done by restricting the computations to areas only where pollution is noticeable. We thus propose to exploit $\mathbf{\Lambda_n}$ as a predictor by turning this into a *binary* weighting mask in the spatial domain. We propose to combine three different models for prediction of pollution visibility: the threshold-vs-intensity (TVI), visual masking and saliency. We will shortly describe each model and how they are combined into a single prediction.

Threshold-vs-Intensity The TVI describes the minimum contrast required to distinguish between foreground and background intensities. A per-pixel test is performed to check whether the residual $\psi(\mathbf{x}) - \mathbf{x_d}$, averaged over some area, is above a threshold:

$$\delta(\mathbf{x}) = (\psi(\mathbf{x}) - \mathbf{x_d}) > \Delta_{TVI}. \tag{7.15}$$

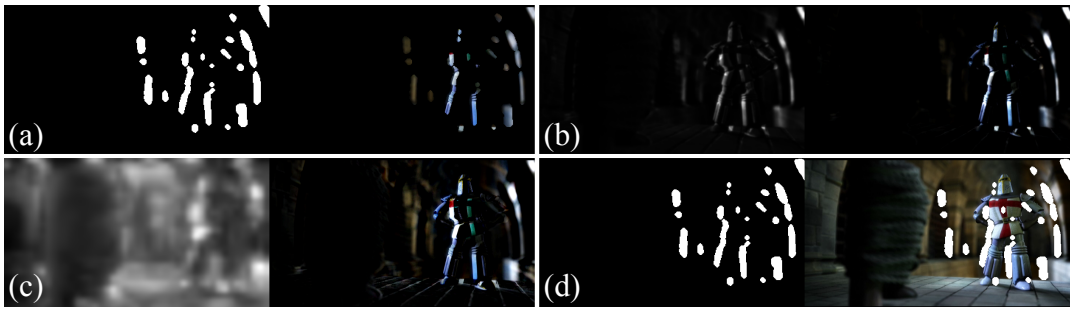


Figure 7.11: Ghosting Prediction Map. *Example of the ghosting prediction map. For each of the four images in this figure the computed map is shown on the left-hand side, and the modulation of the map with the residual is shown on the right-hand side:* ***(a)*** *TVI map,* ***(b)*** *visual masking map,* ***(c)*** *saliency map, and* ***(d)*** *thresholded prediction map.*

Since we are primarily interested in cinema applications with mesopic luminance levels we blend between values computed by photopic and scotopic models [Ferwerda et al., 1996].

Visual Masking Mechanisms in the visual system are tuned to different frequency and orientations bands, and visual masking describes the reduction in contrast sensitivity due to interactions between image components within mechanism bands. We use the model proposed by Ferwerda et al. [1997] with $\bar{\mathbf{x}}$ and $\psi(\mathbf{x})$ as the reference and test images to compute per-pixel masking values:

$$v(\mathbf{x}) = \sum_i \sum_\theta \Delta R^2_{i,\theta}(\mathbf{x}), \tag{7.16}$$

where $\Delta R_{i,\theta}$ is the difference in response of a mechanism with frequency band i and orientation θ, to a reference and test image.

Saliency The predictor can be further extended by considering only ghosting in visually important, or salient, regions, e.g., Harel et al.[2007] proposed an MRF-based approach to predict salient object regions.

To determine the spatial domain weighting mask $\mathbf{\Lambda_n}$ we normalize and combine maps $\delta(\mathbf{x})$, $v(\mathbf{x})$ and $\gamma(\mathbf{x})$ using component-wise multiplication:

$$\mathbf{\Lambda_n} = \gamma(\mathbf{x}) \odot (\delta(\mathbf{x}) \odot v(\mathbf{x})). \tag{7.17}$$

To turn $\mathbf{\Lambda_n}$ into a binary mask we compare each pixel against a threshold $t_{\mathbf{\Lambda_n}}$.

Resolution	#FFTs	#Iterations	Runtime (secs)
2028×1080	3772	159	248.5557
534×844	3992	167	128.4280
435×505	3684	154	34.3573
196×232	2636	109	4.1634

Table 7.1: *Performance of CUDA implementation.*

With deghosting, the pollution is due entirely to unintended light contribution from the *other* eye image. To better predict the noticeability of the ghosting we use the saliency of that eye's image. Thus, for the left eye we have:

$$\mathbf{\Lambda_n}^{\text{left}} = \gamma(\mathbf{x})_{\text{right}} \odot (\delta(\mathbf{x})_{\text{left}} \odot v(\mathbf{x})_{\text{left}}), \tag{7.18}$$

and similarly for $\mathbf{\Lambda_n}^{\text{right}}$.

We implemented this prediction model for our experimental setup. We extracted a 1D CSF from the generated 2D CSF in Section 7.5, and this CSF is used in the masking model. Adaptation luminance for the TVI is computed over a small area as proposed by Ramasubramaniam et al. [1999] (using the *XYZ* measurements obtained for our experimental setup). Figure 7.11 shows an example of ghosting prediction, with $t_{\mathbf{\Lambda_n}} = 0.05$.

Table 7.1 shows the timing results for our algorithm implemented in CUDA. The first row shows the performance for a full resolution (2K) input image. Subsequent rows show performance for examples of areas determined by the pollution prediction. As compensation is only required for these areas, the runtimes for smaller areas greatly reduce. Even with prediction, Table 7.1 shows that our method is not suitable for time-critical applications, but rather for applications which allow the compensation to be performed in an offline preprocess.

7.7 Discussion

We have proposed to incorporate models for properties of the human visual system in the compensation of light pollution in display systems. We address the problem of ghosting in stereoscopic 3D displays in particular: a dim copy of the image intended for one eye is visible for the other eye. This results in a loss of contrast, but more importantly ghosting introduces conflicting edge cues which could hinder the interpretation of depth for an observer. Previous

approaches rely on subtractive compensation, i.e. subtraction of the amount of expected ghosting prior to display. Subtraction could result in negative values if the ghosting occurs in areas of the source image with low intensity. These negative values are clamped to zero, and consequently residual ghosting remains after compensation. We specifically address this problem, and propose to distribute the ghosting smoothly in an locally surrounding area, such that ghosting edges are no longer perceptible. Our approach achieves this by formulating the problem as an optimization, and incorporate the contrast sensitivity function as a weighting. Evaluation shows that our method is preferred to make the stereoscopic viewing experience more comfortable.

Our perceptual compensation method can be applied to additive light pollution, which includes ghosting, in general. We apply perceptual compensation to scattering in immersive display systems. Our results show that perceptual compensation can retain more details compared to previous approaches.

Computational complexity for our optimization-based approach is high due to the fact that $\mathbf{\Lambda_l}$ is dense. Prediction of the areas where ghosting is visible helps to reduce the size of the problem and increase performance. In addition, performance may be improved by formulating an approximation to the perceptual metrics with more desirable properties for optimization. For example an approximation of the CSF kernel which results in a $\mathbf{\Lambda_l}$ that is sparse.

The combination of perceptual models for the non-linear perceptual term $\mathbf{\Lambda_n}$ (Section 7.6), and the subsequent usage of $\mathbf{\Lambda_n}$ to predict when ghosting is above the threshold of perception, was only validated experimentally. For the ghosting prediction to be truly effective, a more thorough understanding of discomfort and fatigue for viewing stereoscopic imagery will be necessary. An example of recent work in this direction explores acceptability thresholds for ghosting [Wang et al., 2011]. However, camera or object motion, stereoscopic saliency, overall scene composition, and disparity gradients influence our contrast sensitivity and depth perception. Further research in these areas will help to develop better computational models for stereoscopic applications.

Compensation for our method is computed based on measurements for a sweet-spot location. However, measurements in a cinema show an increase in ghosting from approximately 1.1% at the sweet-spot, to approximately 2.0% at the periphery. We evaluated the effect of an increase in ghosting for off-axis periphery locations. Figure 7.12 compares subtractive and perceptual compensation for the left periphery, the sweet-spot and the right periphery. The same compensation, based on measurements for the sweet-spot, is applied in all three cases. An increase in ghosting for both periphery locations

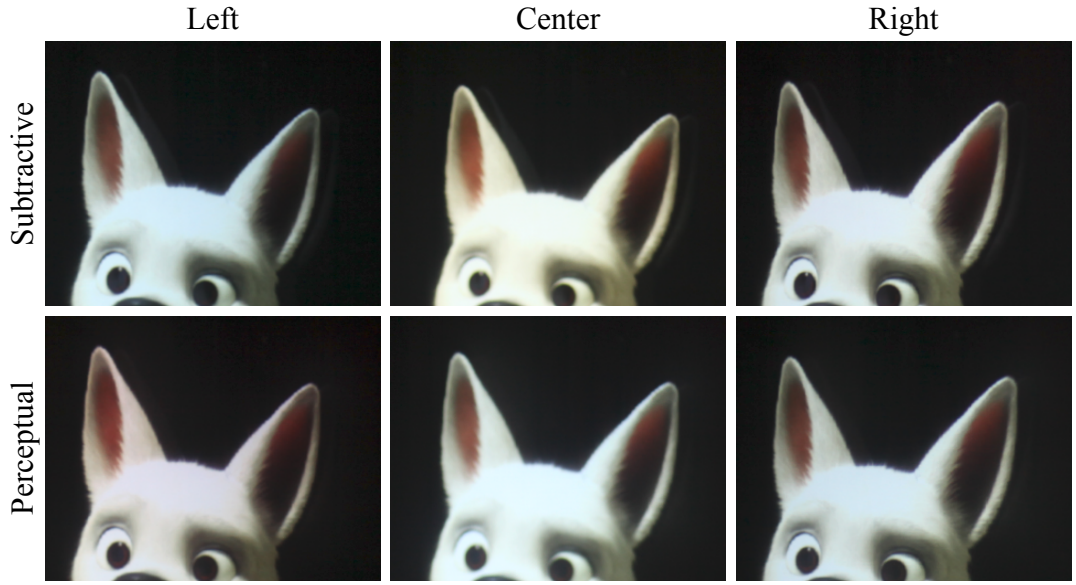


Figure 7.12: Sweet Spot. *We compare compensations for left off-axis, center (sweet-spot) and right off-axis locations. The top and bottom rows compare subtractive and perceptual compensations. Although the ghosting contribution increases for off-axis locations, the perceptual compensation still increases viewing comfort.*

can be observed, however the off-axis locations still benefit from the perceptual compensation.

Cinemas are controlled illumination environments and SMPTE D-Cinema specifications [DCI, 2008] were developed to ensure uniformity among digital cinemas. Therefore, offline pre-computed deghosting material would be valid for all digital cinemas that adhere to the specs. However, stereoscopic 3D content can now be viewed on a variety of consumer devices as well, and the illumination environment will in general not be controlled. All stereoscopic display devices of a specific class, e.g. based on polarizing filters, will likely exhibit a similar amount of ghosting. Given the results above of the experiment for peripheral locations, perceptual deghosting could be applied in non-controlled illumination environments, but only if the computational complexity is reduced to allow the compensation to be computed in real-time. In general, as the exposure to stereoscopic 3D content will increase, ensuring a comfortable viewing experience will become even more important.

CHAPTER 8

Conclusions

8.1 Discussion

Compared to *regular* 2D content, stereoscopic 3D content requires separate images for the left and right eye. However, the difference is more than simply an additional 2D image. For processing, editing, and display of stereoscopic 3D content we have to take the depth of the scene into account as well. In this thesis we have described our research to support and improve the acquisition, processing, editing, and display of stereoscopic 3D content.

The contributions made in this thesis can be summarized as:

- **Acquisition system based on a single reference camera, supported by multi-modal satellite sensors, for computing depth maps and segmentation**
- **Fusion of multi-modal sensor information to compute depth maps using a local method.**
- **Interactive video segmentation approach using multi-modal sensor information. The result of the segmentation is used for computing improved depth maps.**

- **A framework for copy and paste editing of stereoscopic 3D content using depth and segmentation information.**
- **A perceptually-based framework for the compensation of light pollution due to ghosting in stereoscopic 3D displays. The framework is general and can be applied to all forms of additive light pollution in display systems.**

Cinematographers and camera operators are used to capture with a single camera. We therefore propose a multi-modal capture system, using a central high quality reference camera augmented with different types of sensors to support the computation of depth maps and segmentation. Our prototype system demonstrates that it is relatively straightforward to build such a system, including performing geometric calibration and color calibration.

We describe a local method based on fusion of the different modalities for computing depth maps. Depth maps are computed for the high quality reference camera in the capture system. Experimental results were shown for scenes with dynamic objects and background clutter. Occlusions, textureless regions, repeated textures, or similarly colored fore- and background objects may pose problems in methods that rely only on color consistency. Multiple satellite cameras allow us to better estimate occlusion regions by comparing the color consistency between the reference camera and the satellite cameras on the left side, to the color consistency between the reference camera and the satellite cameras on the right side. The fusion of stereo with Time-of-Flight depth data results in the correct reconstruction of textureless regions such as background walls. In addition, surfaces of the same color, but overlapping at different depths can be correctly reconstructed. Of particular interest is the case where human subjects or body parts are overlapping. We showed that different subjects may have different thermal signatures. Therefore, by also fusing the thermal data, an occluding contour can be found even though the skin color is similar. We compared the cases of fusion of stereo with only the Time-of-Flight depth data, fusion with only the thermal data, and fusion with both Time-of-Flight depth and thermal data. Although each of these modalities separately can help improve the depth map, the combination of both gives the best result.

A key challenge in computing depth maps is the estimation of occlusion areas in a scene. Using multiple satellite cameras on either side of a reference camera, helps to better estimate occlusions. To reduce cost and physical footprint of the acquisition system, we propose to use lower quality satellite cameras. However, lower quality cameras exhibit more noise than the high quality reference camera. This is particularly true in low light areas. In addition, the satellite and reference cameras also have very different color spaces. These

properties affect the accuracy of the color consistency between the satellite cameras and the reference camera, and therefore the overall accuracy as well. Computing depth based on color consistency alone will always suffer from ambiguities. Fusion with additional modalities is therefore a promising direction to help solve some of these ambiguities. The Time-of-Flight depth camera resolution is very low compared to the reference camera. Fine details, such as the leaves of a plant, are therefore not accurately captured with the Time-of-Flight depth camera. Fusion with low resolution Time-of-Flight depth thus works best for areas without fine details. Thermal images are most useful when thermal contrast is sufficiently high. This is typically the case for scenes with human actors.

We describe an interactive video segmentation approach to segment multiple foreground objects from the background. Our approach propagates known segmentations for the first and last frame to the intermediate frames in a video sequence. The propagation relies on the matching of superpixels across the video sequence, without any assumption on the motions in a scene. Our method can thus handle moving cameras and non-rigidly moving objects. Exploiting multiple modalities helps to make the matching of superpixels between frames of a sequence more robust. The propagation of known segmentations can handle occluding objects. Provided that foreground objects within a sequence are present in both the first and last frame, they may then disappear and re-appear for the intermediate frames. If optical flow information is available, it can be easily incorporated for the matching of superpixels.

A fully automated method may produce the wrong segmentation. We thus propose to employ a user to interactively correct a propagated segmentation labeling. We require corrections only at a coarse level, rather than at the pixel level, which reduces the burden on the user. Accurate segment boundaries are produced in a subsequent refinement step. Multiple modalities can then be exploited to help resolve color ambiguities and result in better refinement boundaries. The segment boundaries are temporally stable as they accurately match the object boundaries in the video. We can use the boundaries as constraints in the computation of depth maps, so that the depth silhouettes become temporally more stable as well.

We describe an end-to-end system for 3D copy & paste, which extends 2D copy & paste for still images to stereoscopic 3D. The reconstruction of the depth map for the scene is the fundamental operation in this system. The reconstructed depth maps can be used when performing the interactive segmentation, for the propagation of the segmentation result for one eye image to the other eye image, and for composition of the segmented objects into the target scenes. Segmentation, propagation, and composition will all ben-

efit from higher quality depth maps. Direct composition based on the depth map on the other hand, would require an error-free depth map. Error-free depth maps are rarely obtained for general scenes however. Compositing based on depth maps with errors can instead be done using proxy geometry and parametric warps.

When compositing under different orientation, or into a target scene with different stereo parameters, disocclusions could occur. For realistic results these disocclusions would have to be inpainted. Instead we show that by applying the appropriate constraints to compute the parametric warps, disocclusions can be avoided altogether, while still achieving compelling results.

We describe a framework for the compensation of ghosting and scattering. By formulating the compensation as an optimization problem, we can apply the framework to additive light pollution in general. As such, our formulation is a generalization of existing subtractive compensation methods. Since we are compensating for human observers, we should exploit the properties of the human visual system. We show how we can incorporate perceptually-based metrics into the optimization formulation. Specifically, by incorporating the Contrast Sensitivity Function and solving the resulting optimization problem, the residual error is distributed to regions where the human visual system is less sensitive to them. Most importantly, the perceptibility of possibly conflicting edge cues for stereopsis is reduced for perceptual-based deghosting. This makes watching stereoscopic 3D displays more comfortable. A user study was conducted to verify that our perceptually-based compensation method is indeed generally preferred over straightforward subtractive compensation.

Our proposed perceptually-based compensation is a computationally intensive method due to the fact that it is a dense problem. We propose to exploit additional perceptual models for computing a prediction of the visibility of the light pollution. This prediction can then be used to select smaller areas in the images, and apply the compensation on these smaller areas and increase the performance. However, in the case when large areas of the image are impacted by ghosting, running times may still be relatively long.

8.2 Future Work

Based on the research presented in this thesis, we identify several avenues for future work related to acquisition, processing, editing, and display of stereoscopic 3D content.

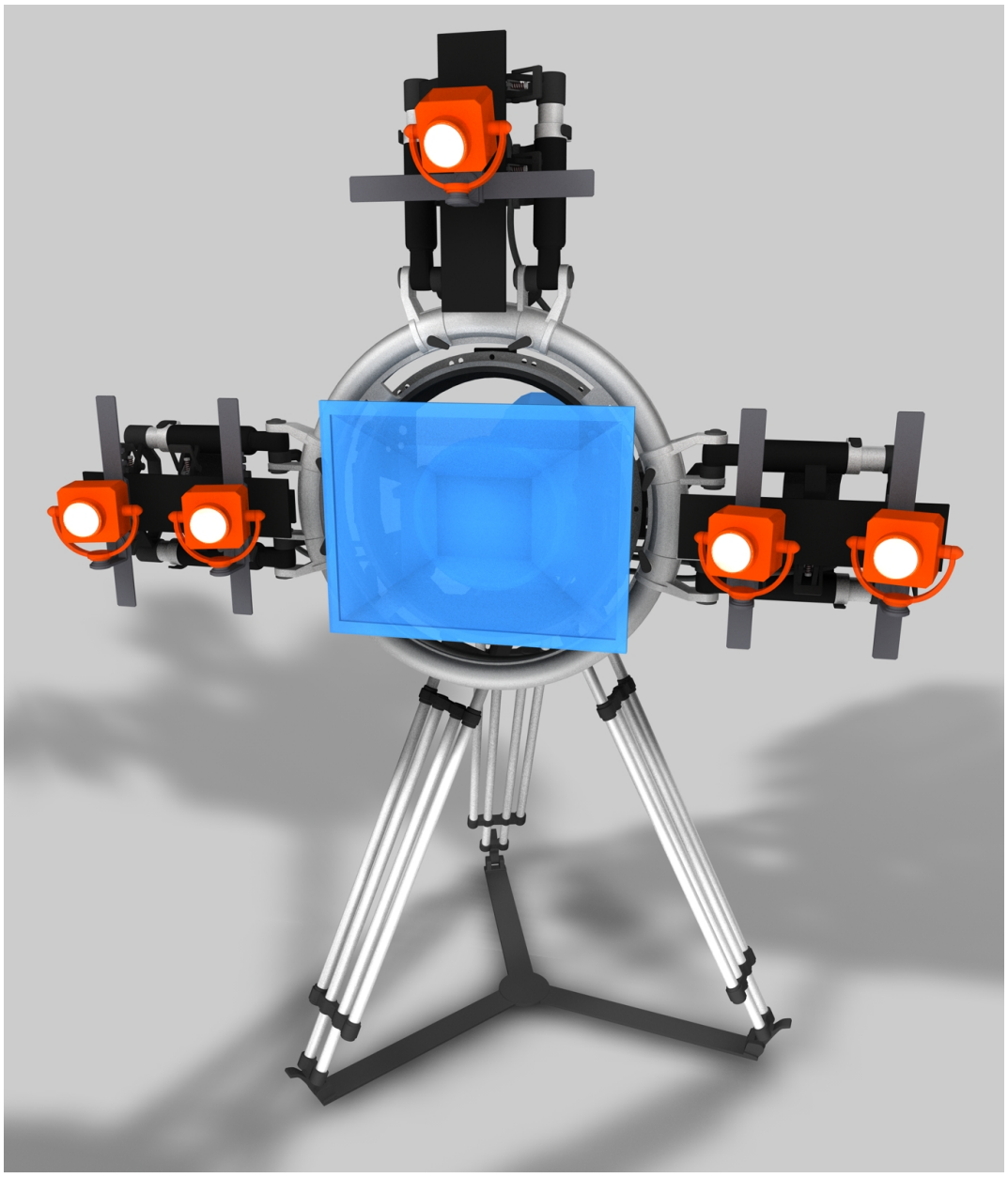


Figure 8.1: Production System. *Concept drawing of a flexible, reconfigurable setup of a high quality reference camera with satellite sensors.*

With respect to the multi-modal capture system we propose, it is important that sensors are mounted rigidly to avoid movement during acquisition. A production ready system should be well-engineered to avoid the sensors from moving. As sensors continue to improve and become smaller we envisage a system where satellite sensors can be easily mounted and reconfigured. Figure 8.1 shows a concept design of a production ready system.

For high quality depth maps, relying on color information alone is not sufficient. As demonstrated, additional modalities can provide cues for computing depth, and also for segmentation. In addition to multiple modalities, for depth maps of challenging scenes to achieve the quality required for movies and broadcast, user interaction will continue to be necessary. An interesting direction for future work would be to understand at which location, how much detail in the depth map is necessary, given the goal of displaying the content on stereoscopic 3D displays. Thin structures remain challenging, but accurate per-pixel depth may not always be required, for example for plants in the background.

An initial step to compute temporally smooth depth maps for video sequences was done by exploiting segmentation information for the video sequence. The segmentation information provides mostly information about the boundaries of foreground objects. The background in the computed depth maps may therefore still exhibit temporally noisy depth values. Combining explicit foreground objects' segmentation information with recent methods proposed by Lang et al. [2012] and Yang et al. [2012b] could address this.

We have discussed a framework for stereoscopic copy & paste editing. Future work should address additional editing operating for stereoscopic 3D content. Furthermore, being able to edit video sequences, rather than single (still) images, will be challenging future work. Temporally varying depth, in combination with varying stereoscopic camera parameters, e.g. baseline, will need to be taken into account for the editing of stereoscopic video sequences.

Perceptually-based deghosting relies on solving an optimization problem, which is computationally expensive. To solve the optimization more efficiently, we would have to investigate how we can formulate the perceptually-based metrics in such a way that are more amenable to optimization. For example, could we formulate an approximation to the Contrast Sensitivity Function which would turn the problem into a sparse system that can be solved more efficiently? In general, to ensure an optimal viewing of stereoscopic 3D content across a wide variety of screen sizes, incorporating perceptual metrics such that stereoscopic 3D content can be automatically adapted to the screen, will be an interesting direction for future work.

APPENDIX A

Depth Maps using Active Illumination and Multi-Spectral Cameras

In this appendix we describe an additional acquisition system using multi-spectral cameras and structured light, plus a method for computing depth maps from the acquired images. The acquisition system resulted from the desire to evaluate the methods presented in Chapters 4 and 5 by comparing the results with ground-truth data. The acquisition system we implemented did not produce the high quality results required to serve as ground-truth data. However, the depth maps that are generated with this system can be considered as an additional multi-modal method for computing depth maps.

A.1 Motivation

Comparison with ground-truth data requires acquisition of a scene with our experimental system, as well as with a system capable of generating ground-truth results. Laser scanners and structured light approaches offer accurate 3D reconstruction. However, these approaches require long scanning times

and are unable to handle dynamic scenes. Dynamic scenes impose more strict requirements on the acquisition. The same scene should be captured in real-time and *simultaneously* with both our experimental system as well as a ground-truth acquisition system. Approaches that are slow, or which interfere with visible illumination are therefore not suitable.

Recently introduced infrared structured light depth sensors [Microsoft, 2012] satisfy both constraints of real-time scanning and avoiding interference with visible illumination. Kinect sensors can obtain depth maps of a scene at 30 frames per second. The reconstruction from a single depth sensor can be very noisy, however multiple depth sensors can be combined instead [Butler et al., 2012, Maimone and Fuchs, 2012]. Although this improves the reconstruction to some degree, the quality is not sufficient for ground truth data. Furthermore, a practical issue is the fact that the different sensors cannot be synchronized to one another, or with external devices. We thus only exploit the ability of the sensors to project infrared speckle patterns, and combine several sensors to obtain a dense coverage of the scene with infrared speckles. Next, we first describe the acquisition system and calibration, followed by a description of our approach for computing depth maps from data captured with the proposed system.

A.2 Acquisition

Figure A.1 depicts the hardware setup. We use the infrared speckle pattern projected by Kinect sensors. The speckle patterns are generated using an infrared laser and a diffraction pattern. Due to the absence of optics the speckle pattern is in focus across the entire working volume. The coverage with speckles is relatively sparse, however the Kinect sensor exploits known locations of the speckles to produce a depth map in real-time. We use multiple sensors in order to densely cover the scene with infrared speckles, without exploiting knowledge about speckle locations.

The infrared speckle patterns can be captured with a camera that is sensitive in the infrared spectrum, and which blocks all visible light contributions. We record the scene with two JAI AD 130-GE multi-spectral cameras, configured as a stereoscopic pair. The cameras uses a dichroic prism to separate the incoming light into visible wavelengths and infrared wavelengths. The camera contains two separate sensors: one which records RGB color images for the visible wavelengths, and one which records grayscale images for the infrared wavelengths. The sensors are pre-aligned such that the image is acquired from nearly the same center of projection. The registered RGB color

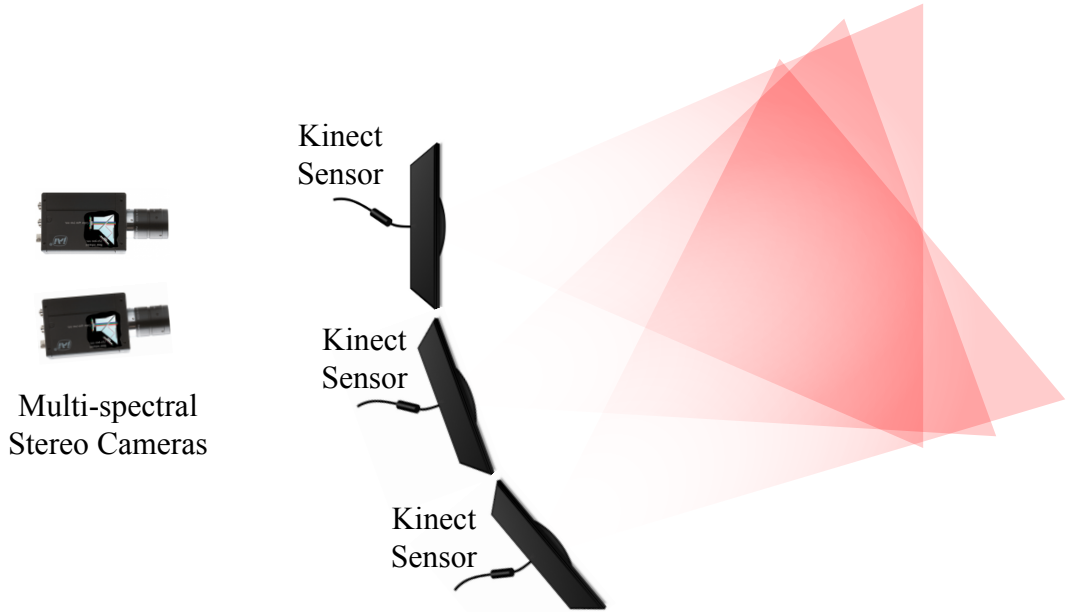


Figure A.1: Acquisition for Validation. *The system consists of a stereo pair of multi-spectral cameras and three Kinect sensors. The Kinect sensors are solely used for projecting infrared patterns generated with the built-in infrared laser and diffraction grating. The projection volumes are overlapped to generate an area of increased speckle density to augment the scene. The multi-spectral cameras contain separate sensors to simultaneously capture RGB images for the visible spectrum and grayscale images for the near infrared spectrum. The sensors are pre-aligned to capture from the same center-of-projection.*

images will be exploited for computing depth maps, as we will explain in Section A.3. Figure A.2 shows an example of the RGB color image and grayscale infrared image that are acquired simultaneously.

The cameras are synchronized to each other, as well as to the sensors of our experimental acquisition system from Chapter 3. We use the same synchronization board as described in Section 3.2.6 for this purpose. The signals are constructed such that the framerate is 25 fps. Note that since the projected speckle pattern is static, there is no need for additional synchronization with the Kinect sensors.



Figure A.2: Multi-Spectral Acquisition. *Example of images acquired simultaneously with the multi-spectral cameras. An internal beam splitting prism splits the light into visible wavelengths and infrared wavelengths.* ***Left*** *RGB color image.* ***Right*** *Infrared grayscale image.*

A.2.1 Calibration

The calibration procedure for the extrinsics and intrinsics is the same as for the satellite cameras of the experimental system described in Chapter 3. We compared the images for the visible spectrum and the near infrared spectrum and found that they are well aligned in the central regions, but some misalignment can be observed towards the periphery of the images. Since we would like to exploit photometric discontinuities in the color images, the color and infrared images should be accurately aligned across the entire image. To correct for the misalignment, we first correct the acquired images for lens distortion. We then compute a homography between the images from the two sensors and warp the RGB color image using this homography. Figure A.3 shows the result before and after alignment using the homography warping.

A.3 Stereo from Multi-Spectral Camera Pair

At every time instance the multi-spectral camera stereo pair produces four images: an RGB and infrared image pair for both the left camera and the right camera, denoted by I_L^{RGB}, I_L^{IR}, I_R^{RGB}, and I_R^{IR} respectively. To compute the depth map we use the Semi Global Matching (SGM) method [Hirschmüller,

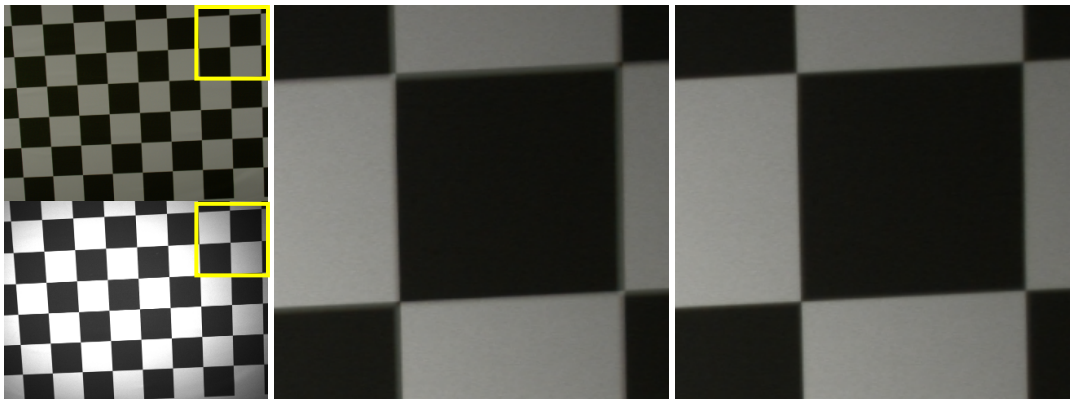


Figure A.3: Multi-spectral Sensor Alignment. ***Left*** *Images for visible spectrum (top) and near infrared spectrum (bottom) sensors. The images are superimposed and zoomed in for the region marked with the yellow rectangle.* ***Middle*** *Before warping the superimposed images appear blurred due to the misalignment.* ***Right*** *After warping the images are correctly aligned and edges in the superimposed image appear sharp.*

2008]:

$$E(D) = \sum_{p} C(p, D_p) + \sum_{q \in \mathcal{N}_p} P_1 \cdot T[|D_p - D_q| = 1] + \sum_{q \in \mathcal{N}_p} P_2 \cdot T[|D_p - D_q| > 1]. \tag{A.1}$$

Here, $C(p, D_p)$ represents the disparity space image computed between the *base* and *match* image [Scharstein and Szeliski, 2002]. Parameters P_1 and P_2 control the amount of smoothness that can be enforced. Parameter P_1 penalizes disparities between neighboring pixels which differ by 1, while P_2 penalizes larger disparity differences between neighboring pixels. The goal is to minimize Equation A.1. For a given base image, SGM approximates the minimization of Equation A.1 by aggregating the cost along multiple paths, each with a different direction, and the minimum cost over all paths is then selected at each pixel. Usually 8 or 16 different directions are considered.

We compute the disparity map $D_{l \to r}$ with the left as the base image, and the right image as the match image. We also compute the disparity map between the right and left image, $D_{r \to l}$. In our case the input to the SGM method is the pair of rectified infrared images I_L^{IR} and I_R^{IR}. The cost $C(p, d)$ at pixel p for disparity d is computed using the dissimilarity measure from [Birchfield and Tomasi, 1998]. When computing $D_{l \to r}$, and $D_{r \to l}$ we can exploit the registered RGB color images I_L^{RGB}, and I_R^{RGB} to make the P_2 penalty value

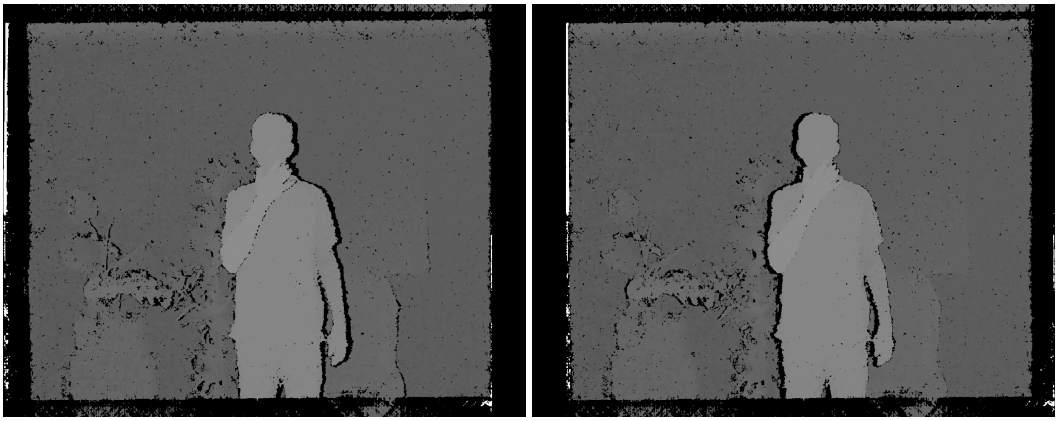


Figure A.4: Left-Right/Right-Left Consistency Check. ***Left*** *Disparity map for left-to-right after consistency check.* ***Right*** *Disparity map for right-to-left after consistency check. Occlusion areas close to the foreground objects are clearly visible.*

spatially varying. Under the assumption that depth discontinuities correlate with color discontinuities, the value of P_2 is weighted by the magnitude of the gradient in either I_L^{RGB}, or I_R^{RGB} at each pixel. This promotes smoothness in uniformly colored areas, while discouraging smoothness near color discontinuities.

Similar to Hirschmüller [2008], we perform a so-called left-right/right-left disparity consistency check [Fusiello et al., 1997]. Disparities which are not consistent, i.e. $|D_{l\to r}(p) - D_{r\to l}(p+d)| > 1$ are set to invalid. An example of resulting disparity maps obtained with out setup is shown in Figure A.4. Note that the disparity consistency check invalidates most disparities in the occlusion regions. Next, we describe how we interpolate invalid disparities.

A.3.1 Interpolating Invalid Disparities

The disparity values which are set to invalid during the disparity consistency check are either noisy or located in occlusion areas. By examining the disparities along the epipolar line in the match image, invalid disparities can be classified as occlusions or mismatches due to noise [Hirschmüller, 2008]. Figure A.5 illustrates the situation. If the epipolar line associated with a pixel in the reference image does not intersect with the disparity surface in the match image, the pixel is classified as occluded. Hirschmüller proposes to interpolate occluded pixels with the smallest disparity value over the different directions. However, interpolation of disparities in the occlusion area

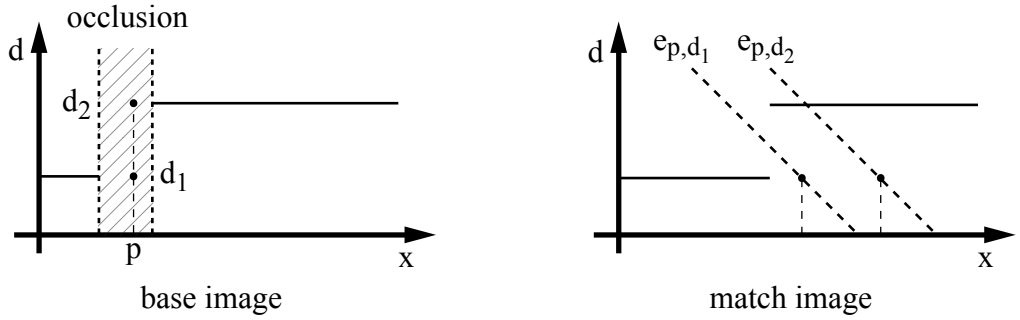


Figure A.5: Occlusion Classification. ***Left*** *Disparities along a scanline for the base image.* ***Right*** *Disparities along the corresponding scanline in the match image. Given a pixel p in the occlusion area, the disparity value can either be d_1 or d_2. The epipolar line e_{p,d_2} intersects the disparity surface in the match image on the right. The epipolar line e_{p,d_1} on the other hand does not intersect the disparity surface.*

requires an accurate occluding contour along the depth discontinuity of an object. Otherwise, the wrong disparity value may be interpolated instead. In our case we obtain strong correlation for the infrared speckles between the base and match image. However, the density of speckles near object boundaries may not be sufficient to obtain accurate silhouettes. Interpolation may then incorrectly assign foreground disparities to pixels in occlusion areas.

Instead, we formulate the interpolation of invalid disparities as an energy minimization:

$$E(d) = \sum_p \phi(x_p) + \sum_{q \in \mathcal{N}_p} \phi(x_p, x_q). \tag{A.2}$$

First, we traverse the image along the same directions as the paths for which minimum costs are computed with SGM. Each path interpolates a possible disparity value. The unary term $\phi(x_p)$ in Equation A.2 is then constructed as follows:

$$\begin{cases} \phi(x_p) = c(p,d) = 0, & \text{if } \texttt{invalid}(p) \;\&\; d \in \mathcal{P}; \\ \phi(x_p) = c(p,d) = 0, & \text{if } \neg\texttt{invalid}(p) \;\&\; d = d_p; \\ \phi(x_p) = c(p,d) = \gamma, & \text{otherwise.} \end{cases} \tag{A.3}$$

Here `invalid()` evaluates to 1 if the pixel is classified as having an invalid disparity, and 0 otherwise. The set $\mathcal{P}$ denotes the set of disparity values obtained from interpolation along the different paths. The value γ denotes a large cost. Equation A.3 ensures that pixels classified as valid retain their current disparity value. On the other hand, if the pixel p has been classified as invalid, and the disparity value d is in the set of interpolated values, the disparities have equal cost of being assigned to the pixel p.



Figure A.6: Disparity Interpolation. ***Left*** *The rectified input image for the left camera.* ***Right*** *The corresponding computed disparity map for the left image. The invalid pixels in Figure A.4-Left are interpolated. The disparities along the depth discontinuity of the foreground object are too noisy for use as ground truth data.*

The binary term $\phi(x_p, x_q)$ in Equation A.2 is a Potts interaction model [Boykov et al., 2001]. Similar as in Section A.3 above for spatially varying weights P_2, we can use the RGB image gradients as weights to make $\phi(x_p, x_q)$ spatially varying. The binary term is then:

$$\phi(x_p, x_q) = w_{p,q}\delta(x_p \neq x_q). \tag{A.4}$$

We solve Equation A.2 using Graph Cuts [Boykov et al., 2001] to interpolate disparities for the pixels classified as invalid. A result is shown in Figure A.6.

A.3.2 Extensions

Although the cameras are synchronized, the response for a speckle in both cameras may be slightly different. This is due to the quantization imposed by the camera in order to produce an image with discrete pixels. The Birchfield-Tomasi (BT) [1998] dissimilarity measure takes this into account for the 1D case. We explored extending the BT dissimilarity to 2D instead. We perform 1D BT dissimilarity for both a horizontal pass c_{horz}, and a vertical pass c_{vert}. The final dissimilarity value is then chosen as $\min(c_{horz}, c_{vert})$. In subsequent experiments we did not observe any significant improvement over using 1D BT dissimilarity.

Another approach would be to combine the cost from the infrared and color images, similar to the fusion described in Chapter 4. The cost volume in

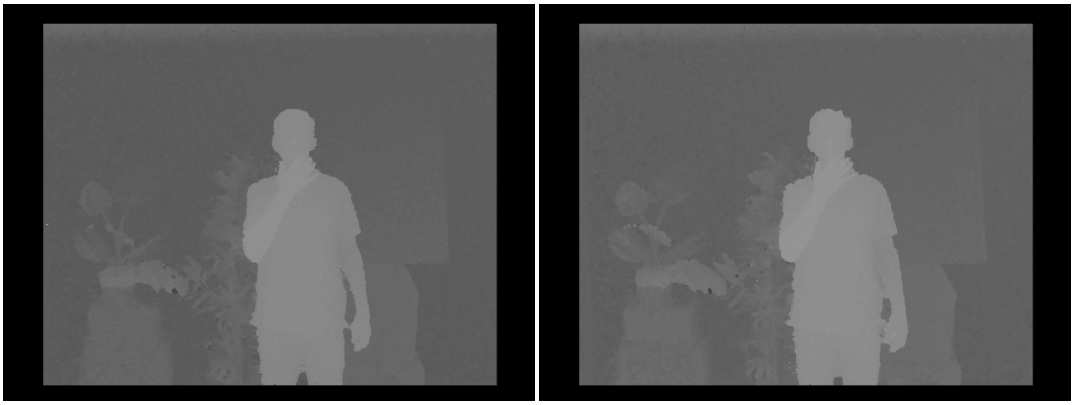


Figure A.7: Infrared vs. Visible and Infrared Fusion. ***Left*** *The result obtained with SGM stereo using correlation on the infrared image only.* ***Right*** *The result obtained with SGM stereo using with fusion of the correlation on visible and infrared images. Overall, using only the infrared image with projected speckle pattern results in better quality disparity maps. Note in particular the area with the arm close to the body, and the area near the top of the head.*

Equation A.1 is then computed as:

$$C(p, D_p) = w_{RGB} \cdot C_{RGB}(p, D_p) + w_{IR} \cdot C_{IR}(p, D_p). \tag{A.5}$$

Compared to the previous result, fusion of the infrared and color images produces results that are inferior in quality. Figure A.7 compares the results for the left image of the stereo pair. The result for SGM stereo using the infrared image only is shown on the left, and the result for stereo using the fusion of RGB color with infrared is shown on the right. In the case of fusion, the depth silhouette on the right-hand side of the person is slightly improved. On the other hand, in the area between the body and the arm, and near the top of the head the results with fusion are worse.

A.4 Comparative Analysis

We calibrate the multi-spectral cameras with respect to the experimental system. This allows us to reproject the depth map obtained for the multi-spectral cameras onto the reference camera of the experimental system. The reprojected result is superimposed on the reference image of the experimental system, shown in the top-right of Figure A.8. The left multi-spectral camera is



Figure A.8: Validation. *Comparison of stereo from projecting IR speckle patterns and sensor fusion. The depth map using IR patterns is obtained using multi-spectral cameras.* ***Top-Left*** *The image from the reference camera of our experimental system (Chapter 3).* ***Top-Right*** *The depth map from Figure A.6 reprojected and superimposed on the reference camera image of the experimental system. Holes (black) are due to disocclusions.* ***Bottom*** *Comparison of the depth map obtained for the multi-spectral cameras (left), with the depth maps obtained for the fusion approach from Chapter 4 (right).*

chosen as the base camera for the multi-spectral cameras stereo pair. We apply a 3×3 median filter to filter holes due to resampling when reprojecting the depth map onto the high resolution reference camera of the experimental system. Some holes (black) remain in areas of disocclusions due to the reprojection.

The reprojected depth map can be compared to the depth maps we obtain with our fusion approach from Chapter 4. The depth map obtained from the multi-spectral stereo camera pair is shown in the bottom-left of Figure A.8, and the depth map obtained from our fusion approach is shown in the bottom-right of Figure A.8. The depth map computed for the multi-spectral cameras contains more noise compared to the depth map obtained with our fusion approach. However, we are only concerned with the compar-

ison of depth discontinuities. As is evident from the result, the boundaries in the depth map computed for the multi-spectral cameras are not accurate enough to qualify as ground truth.

A.5 Discussion

To validate the depth maps we obtain using the multi-modal data acquired with our experimental system, we are especially interested in comparing depth silhouettes for foreground objects. In particular, we would like to compare depths in the case where foreground and background have similar photometric properties. In that case, we expect the Time-of-Flight data and thermal camera to provide the additional information for producing the correct depth boundary.

For computing depth maps using the Kinect speckle patterns and multi-spectral cameras, the infrared speckles alone do not provide sufficiently dense coverage at depth discontinuities. As explained in Section A.3, we have to rely on color discontinuities in the registered RGB images in order to obtain good depth silhouettes. When color discontinuities are sufficiently strong we can achieve good quality depth maps using the Kinect speckle patterns and multi-spectral cameras. However, when foreground and background have similar photometric properties, the resulting depth maps do not have the quality required to be used as ground truth depth data.

Improvements may be obtained by using more sensors to more densely cover the objects. More coverage with infrared speckles also increases the chance that the correlation becomes more ambiguous. The addition of a thermal camera to the setup could help in the case when the photometric properties between the foreground and background are similar. In addition, Kinect sensors may also be positioned 360° around an object to obtain a full 3D reconstruction. Of course, the use of infrared speckle patterns also limits this application to indoor environments.

Bibliography

(2008). Digital cinema system specification v1.2. Digital Cinema Initiatives, LLC, http://www.dcimovies.com.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). Slic superpixels. *Technical Report 149300 EPFL*, (June).

Agarwala, A., Hertzmann, A., Salesin, D. H., and Seitz, S. M. (2004). Keyframe-based tracking for rotoscoping and animation. *ACM Trans. on Graph.*, 23(3):584–591.

Bai, X., Wang, J., Simons, D., and Sapiro, G. (2009). Video snapcut: robust video object cutout using localized classifiers. *ACM Transaction on Graphics*, 28(3):70:1–70:11.

Banham, M. and Katsaggelos, A. (1997). Digital Image Restoration. *IEEE Signal Processing Magazine*, 14(2):24–41.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256.

Bimber, O., Grundhofer, A., Zeidler, T., Danch, D., and Kapakos, P. (2006). Compensating indirect scattering for immersive and semi-immersive projection displays. In *Proc. of the IEEE Conf. on Virtual Reality*, pages 151–158, Washington, DC, USA. IEEE Computer Society.

Bimber, O., Iwai, D., Wetzstein, G., and Grundhöfer, A. (2007). The visual computing of projector-camera systems. In *STAR Proc. of Eurographics 2007*, pages 23–46.

Birchfield, S. and Tomasi, C. (1998). A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406.

Bleyer, M., Rother, C., Kohli, P., Scharstein, D., and Sinha, S. (2011). Object stereo – joint stereo matching and object segmentation. *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Bouguet, J.-Y. (2012). Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/` (Last updated July 9th, 2010). [Accessed 12 September 2012].

Boykov, Y. and Funka-Lea, G. (2006). Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 70(2):109–131.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222 –1239.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In Pajdla, T. and Matas, J., editors, *"European Conference on Computer Vision (ECCV)"*, volume 3024 of *"LNCS"*, pages 25–36, Prague, Czech Republic. Springer.

Butler, D. A., Izadi, S., Hilliges, O., Molyneaux, D., Hodges, S., and Kim, D. (2012). Shake'n'sense: reducing interference for overlapping structured light depth cameras. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 1933–1936, New York, NY, USA. ACM.

CESYS (2012). EFM-01 FPGA Module. `http://www.cesys.com/en/products/kategorie/fpga-boards-spartan/produkt/efm-01-1.html`. [Accessed 01 October 2012].

Chuang, Y.-Y., Agarwala, A., Curless, B., Salesin, D. H., and Szeliski, R. (2002). Video matting of complex scenes. In Hughes, J. F., editor, *Proc. of SIGGRAPH 2002*, pages 243–248. ACM, ACM Press / ACM SIGGRAPH.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 24:603–619.

Conaire, C. O.;Connor, N. E. and Smeaton, A. (2008). Thermo-visual feature fusion for object tracking using multiple spatiogram trackers. *Mach. Vision Appl.*, 19:483–494.

Daly, S. J. (1992). The visible differences predictor: an algorithm for the assessment of image fidelity. In *Human Vision, Visual Processing, and Digital Display III*, volume 1666, pages 2–15. SPIE.

Dehos, J., Zeghers, E., Renaud, C., Rousselle, F., and Sarry, L. (2008). Radiometric compensation for a low-cost immersive projection system. In *Proc. of VR Software and Tech.*, pages 130–133, New York, NY, USA. ACM.

DeMenthon, D. and Megret, R. (2002). Spatio-temporal segmentation of video by hierarchical mean shift analysis. Technical Report LAMP-TR-090,CAR-TR-978,CS-TR-4388,UMIACS-TR-2002-68, University of Maryland, College Park.

Diebel, J. and Thrun, S. (2005). An application of markov random fields to range sensing. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA. MIT Press.

Dong Seon, C. and Figueiredo, M. A. T. (2007). Cosegmentation for image sequences. *Int. Conf. on Image Anal. and Proc.*, pages 635 –640.

Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., and Lischinski, D. (2009). Coordinates for instant image cloning. *ACM Trans. on Graph.*, 28(3).

Felzenszwalb, P. and Huttenlocher, D. (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54.

Ferwerda, J. A., Pattanaik, S. N., Shirley, P., and Greenberg, D. P. (1996). A Model of Visual Adaptation for Realistic Image Synthesis. In *Proc. of SIGGRAPH '96*, Comp. Graphics Proc., Annual Conf. Series. ACM Press / ACM SIGGRAPH.

Ferwerda, J. A., Shirley, P., Pattanaik, S. N., and Greenberg, D. P. (1997). A Model of Visual Masking for Computer Graphics. In *Proc. of SIGGRAPH '97*, Comp. Graphics Proc., Annual Conf. Series. ACM Press / ACM SIGGRAPH.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Fuji (2009). Finepix REAL 3D W3. `http://www.fujifilm.com/products/3d/camera/finepix\_real3dw3/`. [Accessed 5 December 2012].

Fukuda, K., Wilcox, L. M., Allison, R., and Howard, I. P. (2009). A reevaluation of the tolerance to vertical misalignment in stereopsis. *Journal of Vision*, 9(2):1–8.

Fusiello, A., Roberto, V., and Trucco, E. (1997). Efficient stereo with multiple windowing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '97, pages 858–863, Washington, DC, USA. IEEE Computer Society.

Georgiev, T. (2006). Covariant derivatives and vision. *Proc. of European Conf. on Comp. Vision*, 4:56–69.

Gong, M. and Cheng, L. (2011). Foreground segmentation of live videos using locally competing 1svms. *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Grosse, M., Wetzstein, G., Grundhöfer, A., and Bimber, O. (2010). Coded aperture projection. *ACM Trans. Graph.*, 29:22:1–22:12.

Grundmann, M., Kwatra, V., Han, M., and Essa, I. (2010). Efficient hierarchical graph-based video segmentation. *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Guan, L., Franco, J.-S., and Pollefeys, M. (2008). 3d object reconstruction with heterogeneous sensor data. *3DPVT08*.

Harel, J., Koch, C., and Perona, P. (2007). Graph-based visual saliency. In *Advances in Neural Information Processing Systems*, volume 19, Cambridge, MA. MIT Press.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Hawkins, T., Einarsson, P., and Debevec, P. (2005). A dual light stage. *Rendering Techniques*.

Heo, Y. S., Lee, K. M., and Lee, S. U. (2011). Robust stereo matching using adaptive normalized cross-correlation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4).

Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(2):328–341.

Hoffman, D. M., Girshick, A. R., Akeley, K., and Banks, M. S. (2008). Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3).

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

Howard, I. P. and Rogers, B. J. (2002). *Seeing in Depth: Volume 2: Depth perception*. Oxford University Press, New York, USA.

Ilie, A. and Welch, G. (2005). Ensuring color consistency across multiple cameras. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, ICCV '05, pages 1268–1275, Washington, DC, USA. IEEE Computer Society.

Infrared Cameras Inc. (2012). Centurion thermal camera. `http://www.infraredcamerasinc.com`. [Accessed 04 October 2012].

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 559–568, New York, NY, USA. ACM.

Jia, J., Sun, J., Tang, C.-K., and Shum, H.-Y. (2006). Drag-and-drop pasting. *ACM Trans. on Graph.*, 25(3):631–637.

Jorke, H. and Fritz, M. (2003). Infitec - a new stereoscopic visualisation tool by wavelength multiplex imaging. In *Proceedings of Electronic Displays*.

Kajiya, J. T. (1986). The rendering equation. In *Computer Graphics (Proceedings of SIGGRAPH 86)*, volume 13, pages 143–150. ACM.

Klimenko, S., Frolov, P., Nikitina, L., and Nikitin, I. (2003). Crosstalk reduction in passive stereo-projection systems. In Chover, M., Hagen, H., and Tost, D., editors, *Proceedings of Eurographics*, pages 235–240.

Kohli, P., Ladický, L., and Torr, P. (2009). Robust higher order potentials for enforcing label consistency. *International Journal Computer of Vision*, 82(3):302–324.

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1568–1583.

Konrad, J., Lacotte, B., and Dubois, E. (2000). Cancellation of image crosstalk in time-sequential displays of stereoscopic video. In *IEEE Trans. on Image Processing*, volume 9, pages 897–908.

Kooi, F. L. and Toet, A. (2004). Visual comfort of binocular and 3d displays. In *Displays*, volume 25, pages 99–108.

Koppal, S., Zitnick, C., Cohen, M., Kang, S., Ressler, B., and Colburn, A. (2010). A viewer-centric editor for stereoscopic cinema. *IEEE Comp. Graph. and Appl.*, Preprint.

Lalonde, J.-F., Hoiem, D., Efros, A. A., Rother, C., Winn, J., and Criminisi, A. (2007). Photo clip art. *ACM Trans. on Graph.*, 26(3).

Lambooij, M., Fortuin, M., Heynderickx, I., and IJsselsteijn, W. (2009). Visual discomfort and visual fatigue of stereoscopic displays: A review. *Journal of Imaging Science and Technology*, 53(3):30201–1–30201–14.

Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., and Gross, M. (2010). Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. on Graph.*, 29(4).

Lang, M., Wang, O., Aydin, T., Smolic, A., and Gross, M. (2012). Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.*, 31(4):34:1–34:8.

Lantz, E. (1995). Spherical image representation and display: A new paradigm for computer graphics. In Lantz, E., Hutton, M., Savage, S., and Ward, C., editors, *Course #2: Graphics Design and Production for Hemispheric Projection*, ACM SIGGRAPH Course Notes. ACM.

Larsen, E., Mordohai, P., Pollefeys, M., and Fuchs, H. (2006). Simplified belief propagation for multiple view reconstruction. *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 342–349.

Larsen, E., Mordohai, P., Pollefeys, M., and Fuchs, H. (2007). Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8.

Lezama, J., Alahari, K., Sivic, J., and Laptev, I. (2011). Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lindner, M., Lambers, M., and Kolb, A. (2008). Sub-pixel data fusion and edge-enhanced distance refinement for 2d/3d images. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):344–354.

Lipton, L. (2012). Brief history of electronic stereoscopic displays. *Optical Engineering*, 51(2):021103–1–021103–5.

Liu, F., Gleicher, M., Jin, H., and Agarwala, A. (2009a). Content-preserving warps for 3d video stabilization. *ACM Trans. on Graph.*, 28(3).

Liu, J., Sun, J., and Shum, H.-Y. (2009b). Paint selection. *ACM Trans. on Graph.*, 28(3).

Longhurst, P., Debattista, K., and Chalmers, A. (2006). A GPU based Saliency Map for High-Fidelity Selective Rendering. In *Proc. of AFRIGRAPH '06*. ACM.

Loos, B. J. and Sloan, P.-P. (2010). Volumetric obscurance. In *ACM Symp. on Interactive 3D Graph. and Games*, pages 151–156, New York, NY, USA. ACM.

Lourakis, M. A. and Argyros, A. (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30.

Lu, F., Fu, Z., and Robles-Kelly, A. (2007). Efficient graph cuts for multiclass interactive image segmentation. *Proc. of the Asian Conf. on Comp. vision*, pages 134–144.

Lubin, J. (1995). A visual discrimination model for imaging system design and evaluation. In *Vision Models for Target Detection and Recognition*, pages 245–283. World Scientific.

Maimone, A. and Fuchs, H. (2012). Reducing interference between multiple structured light depth sensors using motion. In Coquillart, S., Feiner, S., and Kiyokawa, K., editors, *VR*, pages 51–54. IEEE.

Majumder, A. and Stevens, R. (2005). Perceptual Photometric Seamlessness in Projection-Based Tiled Displays. *ACM Trans. on Graph.*, 24(1):118–139.

Mammen, A. (1989). Transparency and antialiasing algorithms implemented with the virtual pixel maps technique. *IEEE Comp. Graph. and Appl.*, 9(4):43–55.

Mantiuk, R., Daly, S., and Kerofsky, L. (2008). Display adaptive tone mapping. *ACM Trans. Graph.*, 27:68:1–68:10.

Mantiuk, R., Myszkowski, K., and Seidel, H.-P. (2006). A perceptual framework for contrast processing of high dynamic range images. *ACM Trans. Appl. Percept.*, 3(3):286–308.

Microsoft (2012). Kinect. `http://en.wikipedia.org/wiki/Kinect`(Last updated August 27th, 2012). [Accessed 04 October 2012].

Moré, J. J. and Toraldo, G. (1991). On the solution of large quadratic programming problems with bound constraints. *SIAM Journal of Optimization*, 1(1):93–113.

Mukaigawa, Y., Kakinuma, T., and Ohta, Y. (2006). Analytical compensation of inter-reflection for pattern projection. In *Proc. of VR Software and Tech.*, pages 265–268, New York, NY, USA. ACM.

Nadenau, M. J., Reichel, J., and Kunt, M. (2001). Wavelet-based color image compression: Exploiting the contrast sensitivity function. In *IEEE Trans. on Image Proc.*, volume 12, pages 58–70.

Nair, R., Lenzen, F., Meister, S., Schäfer, H., Garbe, C., and Kondermann, D. (2012). High accuracy tof and stereo sensor fusion at interactive rates. In Fusiello, A., Murino, V., and Cucchiara, R., editors, *Computer Vision Ű ECCV 2012. Workshops and Demonstrations*, volume 7584 of *Lecture Notes in Computer Science*, pages 1–11. Springer Berlin Heidelberg.

Ning, J., Zhang, L., Zhang, D., and Wu, C. (2010). Interactive image segmentation by maximal similarity based region merging. *Pattern Recogn.*, 43(2):445–456.

Ochs, P. and Brox, T. (2011). Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. *IEEE International Conference on Computer Vision (ICCV)*.

Paris, S. (2008). Edge-preserving smoothing and mean-shift segmentation of video streams. In *European Conference on Computer Vision (ECCV)*. Springer-Verlag.

Patterson, R. (2007). Human factors of 3-d displays. *Society for Information Display*, 15:861–871.

Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. *ACM Trans. on Graph.*, 22(3):313–318.

Price, B. L., Morse, B. S., and Cohen, S. (2009). Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *IEEE Computer Vision*.

Ramasubramanian, M., Pattanaik, S. N., and Greenberg, D. P. (1999). A Perceptually Based Physical Error Metric for Realistic Image Synthesis. In *Proc. of SIGGRAPH '99*, Comp. Graphics Proc., Annual Conf. Series. ACM Press / ACM SIGGRAPH.

Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *IEEE Comp. Graph. and Appl.*, 21(5):34–41.

Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. (2002). Photographic Tone Reproduction for Digital Images. In Hughes, J. F., editor, *Proceedings*

of SIGGRAPH 2002, Comp. Graph. Proc., Annual Conf. Series, pages 267–276. ACM Press / ACM SIGGRAPH.

Rhee, S.-M., Ziegler, R., Park, J., Naef, M., Gross, M., and Kim, M.-H. (2007). Low-cost telepresence for collaborative virtual environments. *IEEE Trans on Vis. and Comp. Graph.*, 13(1):156–166.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*

Rother, C., Minka, T., Blake, A., and Kolmogorov, V. (2006). Cosegmentation of image pairs by histogram matching. *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 993–1000.

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

Scharstein, D. and Szeliski, R. (2012). Middlebury stereo vision page. `http://vision.middlebury.edu/stereo/` (Last updated April 5, 2012). [Accessed 21 September 2012].

Schuon, S., Theobalt, C., Davis, J., and Thrun, S. (2008). High-quality scanning using time-of-flight depth superresolution. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –7.

Scott, K. (2008). *Planetarium Development Guide*, chapter Theater Configuration. International Planetarium Society.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519 – 528.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2012). Middlebury multi-view stereo vision page. `http://vision.middlebury.edu/mview/` (Last updated April 18, 2012). [Accessed 21 September 2012].

Seitz, S. M., Matsushita, Y., and Kutulakos, K. N. (2005). A theory of inverse light transport. In *ICCV*. IEEE.

Sheng, Y., Yapo, T. C., and Cutler, B. (2010). Global illumination compensation for spatially augmented reality. *Computer Graphics Forum*, 29(2).

Sheskin, D. J. (2007). *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC.

Shum, H. Y., Sun, J., Yamazaki, S., Li, Y., and Tang, C. K. (2004). Pop-up light field: An interactive image-based modeling and rendering system. *ACM Trans. on Graph.*, 23(2):143–162.

Smit, F. A., van Liere, R., and Froehlich, B. (2007). Three extensions to subtractive crosstalk reduction. In *Eurographics Symp. on Virtual Env.*, pages 85–92.

Smith, B., Zhang, L., and Jin, H. (2009). Stereo matching with nonparametric smoothness priors in feature space. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 485–492.

Stone, M. C., Cowan, W. B., and Beatty, J. C. (1988). Color Gamut Mapping and the Printing of Digital Color Images. *ACM Trans. on Graph.*, 7(4).

Sun, J., Zheng, N.-N., and Shum, H.-Y. (2003). Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787 – 800.

Sundstedt, V., Gutierrez, D., Anson, O., Banterle, F., and Chalmers, A. (2007). Perceptual rendering of participating media. *ACM Trans. Appl. Percept.*, 4(3):15.

Szirmay-Kalos, L. (2000). Monte-carlo methods in global illumination. unpublished.

The Foundry (2012). Nuke - Ocula Plug-in. `http://www.thefoundry.co.uk/products/ocula/`. [Accessed 05 December 2012].

Tola, E., Zhang, C., Cai, Q., and Zhang, Z. (2009). Virtual view generation with a hybrid camera array. *EPFL technical report*.

Tsirlin, I., Wilcox, L. M., and Allison, R. S. (2011). The effect of crosstalk on depth magnitude in thin structures. In *Proc. of SPIE Stereoscopic Displays and Applications XXII*, volume 7863.

Vazquez-Reina, A., Avidan, S., Pfister, H., and Miller, E. (2010). Multiple hypothesis video segmentation from superpixel flows. *European Conference on Computer Vision (ECCV)*.

Velodyne (2012). "high definition lidar". `http://velodynelidar.com/lidar/hdlproducts/hdl64e.aspx`. [Accessed 10 December 2012].

Vogel, C. R. (2002). *Computational Methods for Inverse Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Vogiatzis, G., Hernández Esteban, C., Torr, P. H. S., and Cipolla, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2241–2246.

Wang, C. and Sawchuk, A. A. (2008). Disparity manipulation for stereo images and video. *Stereoscopic Disp. and Appl.*, 6803(1):68031E.

Wang, J. and Cohen, M. F. (2008). Image and video matting: A survey. *Foundations and Trends in Comp. Graph. and Vision*, 3(2):97–175.

Wang, L., Jin, H., Yang, R., and Gong, M. (2008). Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE Conf. on Comp. Vision and Pattern Recog.*, pages 1–8.

Wang, L., Teunissen, K., Tu, Y., Chen, L., Zhang, P., Zhang, T., and Heynderickx, I. (2011). Crosstalk evaluation in stereoscopic displays. *J. of Display Technology*, 7(4):208 –214.

Wetzstein, G. and Bimber, O. (2007). Radiometric compensation through inverse light transport. In Alexa, M., Gortler, S. J., and Ju, T., editors, *Pacific Conf. on Comp. Graph. and Appl.*, pages 391–399. IEEE Computer Society.

Wheatstone, C. (1838). Contributions to the physiology of vision - part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. In *Philosophical Transactions*, volume 128, pages 371–394. Royal Society of London.

Wikipedia–3D Film (2013). William friese-greene. `http://en.wikipedia.org/wiki/3D_film` (Last updated January 19th, 2013). [Accessed 22 January 2013].

Yang, Q., Wang, L., Yang, R., Stewenius, H., and Nister, D. (2006). Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2347–2354, Washington, DC, USA. IEEE Computer Society.

Yang, Q., Yang, R., Davis, J., and Nister, D. (2007). Spatial-depth super resolution for range images. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8.

Yang, R. and Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:I–211–I–217 vol. 1.

Yang, W., Zhang, G., Bao, H., Kim, J., and Lee, H. Y. (2012a). Consistent

depth maps recovery from a trinocular video sequence. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1466 –1473.

Yang, W., Zhang, G., Bao, H., Kim, J., and Lee, H. Y. (2012b). Consistent depth maps recovery from a trinocular video sequence. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1466 –1473.

Yeh, Y. and Silverstein, L. (1990). Limits of fusion and depth judgment in stereoscopic color displays. *Human Factors*, 32(1):45–60.

Zhang, G., Jia, J., and Bao, H. (2011). Simultaneous multi-body stereo and segmentation. In *IEEE International Conference on Computer Vision (ICCV)*.

Zhang, G., Jia, J., Wong, T.-T., and Bao, H. (2009). Consistent depth maps recovery from a video sequence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):974–988.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhu, J., Wang, L., Yang, R., and Davis, J. (2008). Fusion of time-of-flight depth and stereo for high accuracy depth maps. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

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

Zitnick, C. L., Jojic, N., and Kang, S. B. (2005). Consistent segmentation for optical flow estimation. *IEEE Int. Conf. on Comp. Vision*, pages 1308–1315.

Zitnick, C. L. and Kang, S. B. (2007b). Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75:49–65.

Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. (2004). High-quality video view interpolation using a layered representation. *ACM Trans. on Graph.*, 23(3):600–608.

Zwicker, M., Pfister, H., van Baar, J., and Gross, M. (2002). Ewa splatting. *IEEE Trans. on Vis. and Comp. Graph.*, 8(3):223–238.