UNIVERSITY OF CALIFORNIA, SAN DIEGO

Revolutionizing Laparoscopy:
Bringing Glasses-free Multiview 3D into the Operating Room

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Ramsin Khoshabeh

Committee in charge:

Professor Truong Q. Nguyen, Chair
Professor Mark A. Talamini, Co-Chair
Professor Pamela C. Cosman
Professor William S. Hodgkiss
Professor Yu-Hwa Lo

2012

The dissertation of Ramsin Khoshabeh is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California, San Diego

2012

DEDICATION

To my father and mother, *Leon and Valentina*, because you showed me the meaning of agape love long before my mind ever conceived of it.

To my grandmother, *Marusa*, because you have endured more than I could ever imagine.

To my brother, *Raman*, because you are the rock of our family.

To my sister, *Ramina*, because you are the heart of our family.

To my sister-in-law, *Ania*, because you hold our family together.

To my nephew, *Sebastian*, because you complete our family and no one screams as loud as you.

Most importantly, to my Lord and Savior, *Jesus Christ*, because the Resurrection is a fact. Life has no meaning apart from You, but in You the means are found to die to myself and genuinely live to love others.

EPIGRAPH

He is no fool who gives what he cannot keep
to gain that which he cannot lose.”
—Jim Elliot, *In the Shadow of the Almighty*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to express my heartfelt appreciation to a dear lady by the name of Stacey Liekweg. She was the catalyst that brought all of this work into existence. There was a moment when I had given up hope in completing my graduate education because of a large number of difficulties, but she compelled me to persevere through them. I am indebted to her more than she will ever know.

I would also like to thank Professor Truong Nguyen as the chair of my committee and Dr. Mark Talamini as the co-chair. Over the years, Professor Nguyen has grown to become like family to me. His guidance and support have been unending, and truthfully he is one of the greatest advisors a student could hope to have. Dr. Talamini has been an inspiration, a source of great wisdom, and without his support, there is no way that this research could have come as far as it has in just a few years.

Next I want to thank everyone in the Video Processing Lab. I have had many great memories with my colleagues here, particularly (in alphabetical order by last name) Can Bal, Stanley Chan, Kris Gibson, Natan Jacobson, Ankit Jain, Jason Juang, and Lam Tran, all of whom have contributed in various degrees.

Additionally, I would like to express heartfelt appreciation to the surgeons and staff of the Center for the Future of Surgery at UCSD, who gave us access to their facilities, aided in setting up the experiments, and participated in the study.

Last, but not least, I must extend my deepest gratitude to Mr. Nathaniel Feldman for proofreading this entire dissertation and providing excellent feedback.

Chapters 4, 5, and 8, in part, are reprints of the material as it appears in [1, 2, 3]. The dissertation author was the primary investigator and author in [1] and secondary in [2, 3].

Chapters 6 and 8, in part, are reprints of the material as it appears in [4, 5]. The dissertation author was a secondary investigator and author in these works.

VITA

2005	B. S. in Electrical and Computer Engineering, *Summa cum Laude*, University of California, San Diego
2005-2007	M. S. in Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California, San Diego
2008-2012	Ph. D. in Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California, San Diego

PUBLICATIONS

Z. Lee, R. Khoshabeh, J. Juang, and T. Q. Nguyen, "Local Stereo Matching using Motion Cue and Modified Census in Video Disparity Estimation," *to appear in European Signal Processing Conference*, 2012.

K. R. Lee, R. Khoshabeh, and T. Q. Nguyen, "Sampling-based Robust Multilateral Filter for Depth Enhancement," *to appear in European Signal Processing Conference*, 2012.

S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Video Restoration," *Transactions on Image Processing*, 20, 3097-3111, 2011.

R. Khoshabeh, S. Chan, and T. Q. Nguyen, "Spatio-temporal Consistency in Video Disparity Estimation, *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An Augmented Lagrangian Method for Video Restoration," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

L. Tran, R. Khoshabeh, A. K. Jain, C. Pal, and T. Q. Nguyen, "Spatially Consistent View Synthesis with Coordinate Alignment," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

A. K. Jain, L. Tran, R. Khoshabeh, and T. Q. Nguyen, "Efficient Stereo-to-Multiview Synthesis," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

R. Khoshabeh, C. Bal, A. K. Jain, L. Tran, S. H. Chan, and T. Q. Nguyen, "Next-generation 3D: From Depth Estimation to the Display," *COMSOC Multimedia Communications Technical Committee E-Letter*, 2011.

R. Khoshabeh and J.D. Hollan, "Spatio-temporal Interest Points for Video Analysis," *International Conference on Human Factors in Computing Systems (CHI)*, 3455-3460, 2009.

M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. D. Hollan, and J. O. Borchers, "SLAP widgets: Bridging the Gap between Virtual and Physical Controls on Tabletops," *International Conference on Human Factors in Computing Systems (CHI)*, 481-490, 2009.

M. Weiss, R. Jennings, R. Khoshabeh, J. O. Borchers, J. Wagner, Y. Jansen, and J. D. Hollan, "SLAP widgets: Bridging the Gap between Virtual and Physical Controls on Tabletops," *CHI Extended Abstracts*, 3229-3234, 2009.

M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. D. Hollan, and J. O. Borchers, "SLAPbook: Tangible Widgets on Multi-touch Tables in Groupware Environments," *International Conference on Tangible and Embedded Interaction (TEI)*, 297-300, 2009.

M. Weiss, R. Jennings, J. Wagner, R. Khoshabeh, J. O. Borchers, and J. D. Hollan, "SLAP: Silicone Illuminated Active Peripherals," *Tabletops and Interactive Surfaces*, 2008.

R. Khoshabeh, T. Gandhi, and M. Trivedi, "Multi-Camera Based Traffic Flow Characterization & Classification," *International Intelligent Transportation Systems Conference*, 2007.

ABSTRACT OF THE DISSERTATION

Revolutionizing Laparoscopy: Bringing Glasses-free Multiview 3D into the Operating Room

by

Ramsin Khoshabeh

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California, San Diego, 2012

Professor Truong Q. Nguyen, Chair
Professor Mark A. Talamini, Co-Chair

Over the past several years, a dramatic increase in attention has been paid to 3D media, particularly in the movie industry. Due to the lack of previous algorithmic advancement and the absence of unobtrusive technology, however, three-dimensional visualization has failed to penetrate the operating room until now. In this work, we present a system to deliver glasses-free 3D visualization of laparoscopic surgeries to the operating room via multiview autostereoscopic displays. We begin by developing robust stereo-to-multiview content generation so that we may produce an arbitrary number of stereo sequences for presentation on autostereoscopic displays. We then introduce a reliable disparity estimation technique that enables the synthesis of the virtual views. In addition, we enforce spatio-temporal consistency in disparity estimates to provide a superior 3D experience for surgeons and operating room staff. Finally, we implement the entire system on graphics hardware in order to achieve real-time operability.

Numerous attempts have been made to introduce three-dimensional (3D) video systems into clinical routine, particularly for surgeries. The drawback with all of them thus far has been the fact that they require users to wear cumbersome glasses in order to receive the advantage of stereoscopy. In this work, we present, to our best knowledge, the world's first laparoscopic surgical system that delivers glasses-free multiview 3D in high-definition (HD) resolution. In addition to a quantitative evaluation of the video processing components, we performed an initial subjective study with laparoscopically experienced surgeons, which yielded very promising results.

Chapter 1

Introduction

Take a stroll through most hospitals and medical research facilities today, and you are bound to come across scores of technological innovations. Some are invisible to the naked eye, such as advanced algorithms to handle deoxyribonucleic acid (DNA) sequencing faster than ever before, while others simply cannot be missed, such as magnetic resonance imaging (MRI) machines. Each technology serves a uniquely different purpose, but, in the broadest sense, they are all utilized in pursuit of the same goal – improving our capabilities of diagnosing, preventing, and treating every known health ailment in the best way possible.

In this thesis, we focus our attention on one subset of the medical world – surgery, and narrow it down even further to the particular practice of laparoscopic surgery. Our aim is to evaluate the practice in its current state and deduce ways that technological advances may ameliorate some of the present weaknesses. In particular, we focus our attention on the fact that, even though laparoscopic surgery has many benefits (*e.g.*, operating in hard-to-reach places like the prostate), it nonetheless possesses intrinsic shortcomings that have yet to be addressed by the technological world.

The goal of laparoscopic surgery is to perform an operation with minimal incisions to the body of the patient. Typically, each incision is on the order of a few millimeters in length. In abdominal laparoscopy, the abdomen is inflated with carbon dioxide gas to create a more spacious working environment by separating the abdominal wall from the internal organs. Surgical implements are inserted through incisions through which a surgeon may perform the operation. Typically, at least one of the incisions is used for a camera system that relays footage to a traditional LCD monitor. The surgeon is left with only the two-dimensional images to perform the operation.

Due to the working environment of these operations, being deprived of the natural haptic and optic senses, surgeons must undergo lengthy training processes in order to become adept at working laparoscopically. In particular, they must be able to navigate a three-dimensional environment using just a two-dimensional representation of it, a difficult and often dangerous task.

In this dissertation, we focus on trying to restore a surgeon's natural depth perception to laparoscopic surgeries. Throughout the chapters, we will discuss the individual components that are required to build such a system, both in hardware and software. In fact, in the final chapters, we show results of our system being used in an initial study by laparoscopically trained surgeons. Our hope is that one day a fully realized system will become standardized in laparoscopic surgeries, and perhaps in even other segments of the medical world.

Chapter 2

Is 3D Really Necessary?

2.1 Motivation

© 2007 RelayHealth and/or its affiliates. All rights reserved.

(a) Illustration of a laparoscopic surgery. (b) Image of the DaVinci Robotic System.

Figure 2.1: Example modern advances in surgery.

It is undeniable that the future of medicine lies intertwined with technology. Even today we see enormous growth in the medical technologies industry. Particularly in the realm of surgery, we observe that over the past few years the operating room has blossomed into a haven for innovative technologies. While most advances have led to enhanced capabilities for surgeons, they also have increased the degree of physical separation between the physician and the patient. For example, in laparoscopic surgeries (a division of minimally-invasive surgeries (MIS)), surgeons can perform intricate operations via surgical implements and a camera inserted through trocars (see Fig. 2.1a), but this comes at the price of diminishing the senses of sight (depth perception) and touch (tactile dexterity).

Without depth perception, simple tasks, such as suturing, become appreciably more difficult and time consuming [6, 7, 8, 9, 10, 11]. Additionally, the risk for injuries increases due to misperceptions of the three-dimensional structure of objects within the human anatomy [12]. In fact, surgeons wishing to perform laparoscopy have to undergo many hours of additional training to be able to comfortably operate in a 3D world (e.g., the abdominal cavity) by seeing only the 2D representation of it on an LCD. This gulf between the patient and the physician is further increased as technological advances now begin to offer surgeons the capability of performing the operation from a completely remote location.

While there are some obvious advantages to operating laparoscopically, such as minimal scarring and faster recovery times, they should not come at the expense of having the physician's natural senses dulled, resulting in increased stress levels [13]. A few of today's most advanced surgical technologies, such as the DaVinci robotic system [14], do in fact offer 3D viewing capabilities. This is accomplished by using a stereoscopic endoscope along with a viewing console where a surgeon sees a separate camera view from each eye, creating a 3D effect. This is of no benefit to anyone else in the operating room (OR), however, since only a single person may view the 3D image. For longer surgeries, this is also not ideal because, in order to see the 3D effect, the surgeon must remain seated in front of the console, which contributes to fatigue and restricts the ability to move around the room as shown in Fig. 2.1b. What is possibly most daunting of all is that these systems are *extremely* expensive, often outside of the budget of most facilities. Nonetheless, we see that 3D is in fact a desirable element of the surgical experience. Others, such as [6], have taken the more traditional approach of using passive filtered glasses with a wavelength-multiplexed monitor to deliver 3D. Their studies reveal that greater depth perception does in fact increase surgeon accuracy in laparoscopic surgeries, but again they are limited to glasses-based systems.

In the past, researchers attempted to solve the problem of lost depth perception by using anaglyph (color multiplexed) images with corresponding glasses so that each eye would perceive a separate image, creating the sensation of three dimensions (3D). While such glasses were gradually popularized in the movie industry, they did not fare well in the medical world, in spite of positive findings such as those of [6]. The approach of using glasses for surgeries has often been unsuccessful because 3D glasses are cumbersome, disorienting, and nauseating with prolonged usage. Furthermore, they are a hazard in the operating room because they may fall off, they are a potential source of contaminants, and the physician is not always looking at just the display. Looking at anything besides the screen causes users eye strain and discomfort and makes them unable to use their visual system for fine-grain activities such as surgery. The same problem would exist for more sophisticated glasses, such as polarized or active shutter glasses.

Despite the indispensability of depth perception, surgeons (and most people, for that matter) generally oppose having to wear any form of eyewear in order to experience 3D. Many would rather suffer the loss of depth if it means that they can perform the operation unhindered. Due to recent technological advances, however, it is no longer necessary for them to have to pick one or the other. Modern multiview autostereoscopic displays allow an individual to perceive high quality 3D without the need for glasses. Along with advanced video processing algorithms (our primary contribution), they have the potential to deliver immersive 3D technology into the operating room to mitigate the negative effects of traditional 2D displays. No other system to date has shown such capabilities.

2.1.1 3D and Surgery

Two-dimensional (2D) displays cannot convey the depth information necessary to perform a remote operation. Even in traditional MIS, surgeons undergo many hours of training to be able to operate in a 3D environment (e.g., the abdominal cavity) by seeing only the 2D representation of it on an LCD. We see from technological advances, such as the DaVinci, that 3D is a desired element of the surgical experience. Furthermore, in a 2002 study done on the causes of laparoscopic bile duct injuries, the experiments revealed that the ”primary cause of error in **97% of cases** was a visual perceptual illusion... this stemmed from an illusion of object form.” (emphasis added) [12]. These results reveal the absolute necessity of having a virtual representation of real-world objects that attempts to maintain their 3D structure as much as possible.

In our work, we aim to deliver immersive 3D technology into the operating room to mitigate the negative effects of traditional 2-dimensional (2D) displays. The approach utilizes autostereoscopic displays to visualize synthesized 3D images that can be seen with the naked eye. In this way, we enable everyone in the OR to see 3D video from the surgical cameras without the need for any special glasses.

Chapter 3

Multiview Autostereoscopic Displays

3.1 Seeing in 3D

Before we discuss multiview autostereoscopic displays and their function, we delve into the Human Visual System (HVS) for a brief background on what it means for a person to actually see in 3D. The HVS is a complex system that integrates a vast number of optical cues, learned associations from memory, and geometric transformations to produce the sensation known as sight. When we refer to seeing in three dimensions, we refer to the discriminability of the human mind to discern the true spatial location of objects in the perceived world through the HVS.

In this general sense, people have been seeing 3D on their television sets since the very first electro-mechanical models were introduced to the public in the 1920s. This is because many cues within the scene led the viewers' minds to infer depth, even though their visual systems were seeing a two-dimensional image (the television screen). Fig. 3.1 illustrates some of these so-called monocular cues that provide humans a sense of depth perception, even when viewing a flat image or observing with just one eye. Artists and painters have long known about these phenomena and continue to exploit them in their art to this day. As the reader

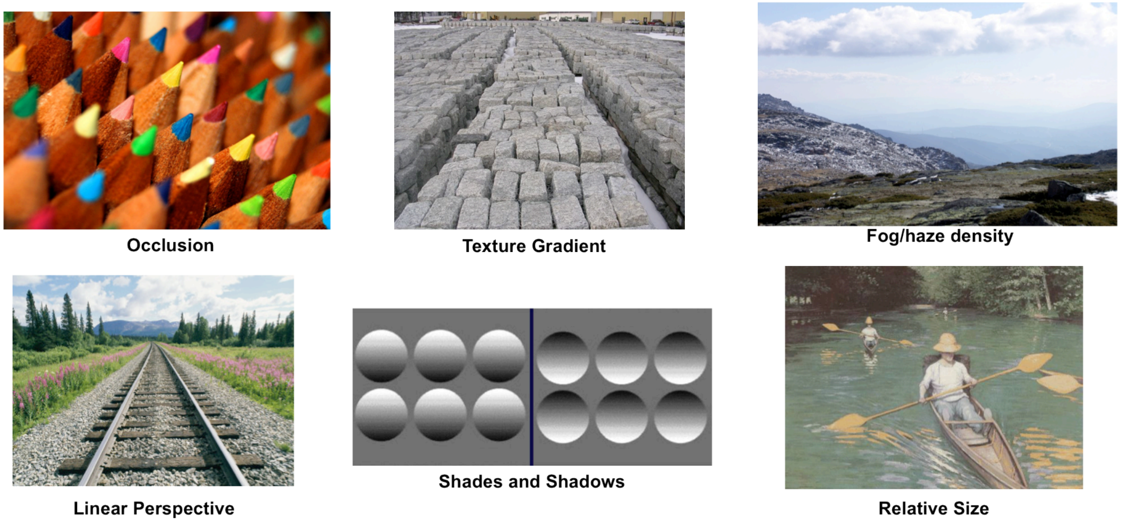


Figure 3.1: Common monocular visual cues used by HVS to perceive depth [15].

will attest, each of the six images provides some form of depth information, even though this is a flat document. For instance, the shrinking of the size of the train tracks in the bottom left image gives the insight that objects are farther from the viewer as you move to the top of the image. However, it is just as reasonable that a flat (no depth) structure might be built with two metal beams converging into an 'A' shape and that would be what we are observing in this figure. Yet, because of our strongly learned association that train tracks must run in parallel, it is almost impossible to break this association and imagine it as a flat structure, even if we try. Therefore, the cue dominates, and we perceive depth in the scene.

Monocular cues are not perfect and can often be used to trick the HVS to see depth when there really is none in the scene, especially in non-natural scenes. As depicted in Fig. 3.2, each of the set of texture patterns gives the sense that there exists a 3D structure in the figure when, in reality, some of the same monocular cues we just mentioned are being used to fool the viewer. For example, the right-most figure is exploiting a texture gradient to give a tunnel-like illusion.

While monocular cues are powerful, it would be almost foolish for a display manufacturer to claim their television set is a 3D system because it provides monocular cues, since they are inherent in any imaged scene. Unfortunately, the term "3D" is overloaded with meanings today and is often misunderstood.

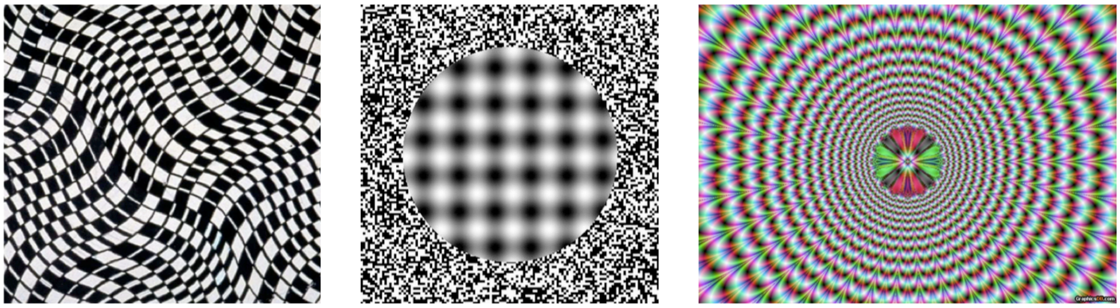


Figure 3.2: Some illusions exploiting monocular visual cues.

When people refer to 3D or 3D systems, they are generally referring to the idea of stereopsis, whether they know it or not. Stereopsis, or binocular vision, is the impression of depth that we perceive because we have two eyes viewing the same scene. Although it has been debated for years, many researchers believe that it is the most powerful visual cue for depth perception. The way that binocular vision works is that, when the eyes are focused at infinity, objects closer to the individual will appear to have a larger horizontal displacement in one eye relative to the other than objects that are further away. This disparity informs the mind of the location of objects in depth. Furthermore, disparity is not completely dependent on monocular cues. A surface could be either textured or flat and still have large disparity, appearing close.

3.2 3D with Glasses

Glasses-based systems offer a solution to 3D visualization by taking advantage of stereopsis. Each eye is shown a separate image just as if an individual was viewing the natural world. This is accomplished by interleaving the stereo views into a single image for the viewer. Many approaches have been proposed over the years on how to appropriately design the glasses and they are well reviewed [16, 17, 18, 19]. Nowadays, the solutions range from simple anaglyphs to the more recent active shutter systems.

Regardless of complexity, however, the glasses always serve the same purpose. Since two camera feeds are interleaved on the display, the goal of the glasses is to split the image into the left and right components. With anaglyph glasses, the images are interleaved in the color bands and then split with filters at particular wavelengths, resulting in color distortion. With passive glasses, the images are interleaved spatially and the scanlines are polarized so that the glasses may filter out the image of the opposite eye. The drawback with passive filtering is a loss in vertical image resolution. Active shutter glasses synchronize with the display system using an infrared connection. When the display shows an image for the left eye, the shutter on the right eye activates, blocking the eye from seeing it. The converse happens for the right eye. At sufficiently high frame rates (typically 60 Hz and above), the eyes each see a separate, continuous video. Yet, with active shutter glasses, there is a dimming effect due to the alternating black frames, a loss of temporal resolution, and the active components mean that the glasses in general cost much more.

The problem with all of these solutions is that viewers must wear glasses to see in 3D. They encumber viewers, cause fatigue, and usually lower the appeal of 3D systems. Furthermore, none of the solutions provides a full 3D experience without degrading the viewing quality in some respect.

3.3 Autostereoscopic Displays

To enable 3D viewing without the need for any glasses, researchers began to seriously explore autostereoscopic displays as early as the 1990s. Autostereoscopic displays do not require observers to wear glasses, eliminating a key obstacle to the mainstream acceptance of 3D displays, but they require significant changes in the design of 3D display systems.

On a broad scale, there exist two types of technologies that are commercially available and are being actively developed – parallax barrier and lenticular systems. Parallax barriers are masks that occlude certain pixels when viewed from a certain location. For a two view parallax barrier, the left and right images are typically interlaced in alternating columns on the LCD. Then the barrier is positioned so that the left and right pixels are blocked from view except when viewed from the left and right eyes' viewing window respectively. Parallax barriers suffer from reduced brightness but are widely used in a number of commercial applications. On the other hand, lenticular systems attempt to achieve the same result as parallax barriers, but instead of blocking certain pixels from sight, they employ optical elements to refract light to the individual eyes. The end result is that the reduced brightness seen in parallax displays does not exist but at the cost of a more complicated system of lenses.

3.3.1 Two-view Lenticular Systems

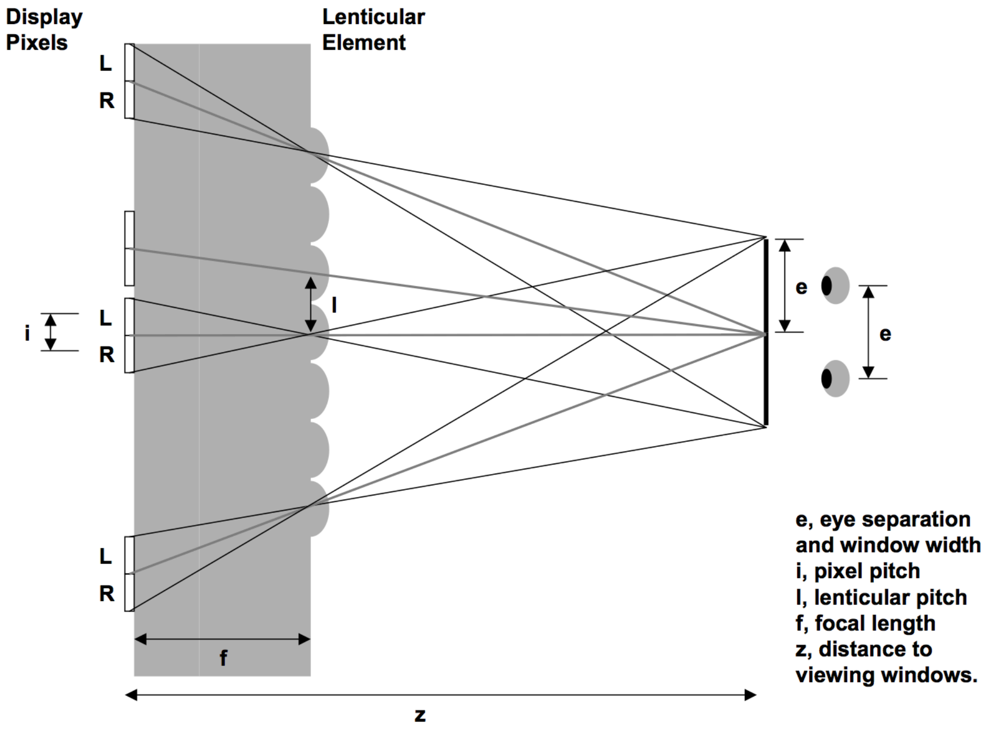


Figure 3.3: A 2-view lenticular autostereoscopic display [20].

A typical two view lenticular system is described in [20] and illustrated in Fig. 3.3. In the figure, a top-down view illustrates the pixels of the display (on the left) being projected by cylindrical lenses to the appropriate viewing window for each eye (on the right). Although not pictured, due to the nature of the optics, these viewing windows repeat along the plane parallel to the display. As long as

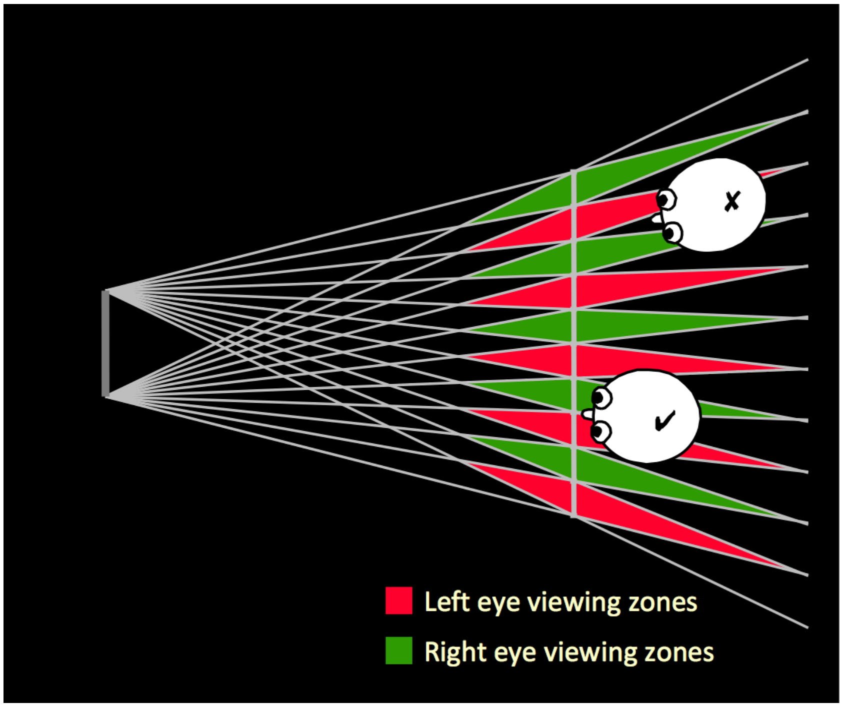


Figure 3.4: A 2-view lenticular screen with upper head in the wrong position [21].

the viewer's left eye remains in the left zone and the right eye remains in the right zone, the 3D sensation will be present. However, if the eyes are in the wrong position, the viewer will see inverted stereo as shown in Fig. 3.4. In this figure, the upper head will be in the wrong place and receive a strange, uncomfortable visual experience.

3.3.2 Multiview Lenticular Systems

To minimize the locations where an observer might receive wrong, inverted stereo and to allow multiple viewing zones for a look-around effect, we can enlarge the lenticular lenses to cover multiple pixels, thereby increasing the number of views. One of the earliest and longest-standing multiview systems is the solution developed by Philips [22], which provides 7 distinct viewing perspectives. Yet, in the early 1990s, it was thought that four views was the theoretical limit that a multiview display could show [21]. This was because the initial multiview approaches all used vertical cylindrical lenslets. So for a 4-view display, the hori-

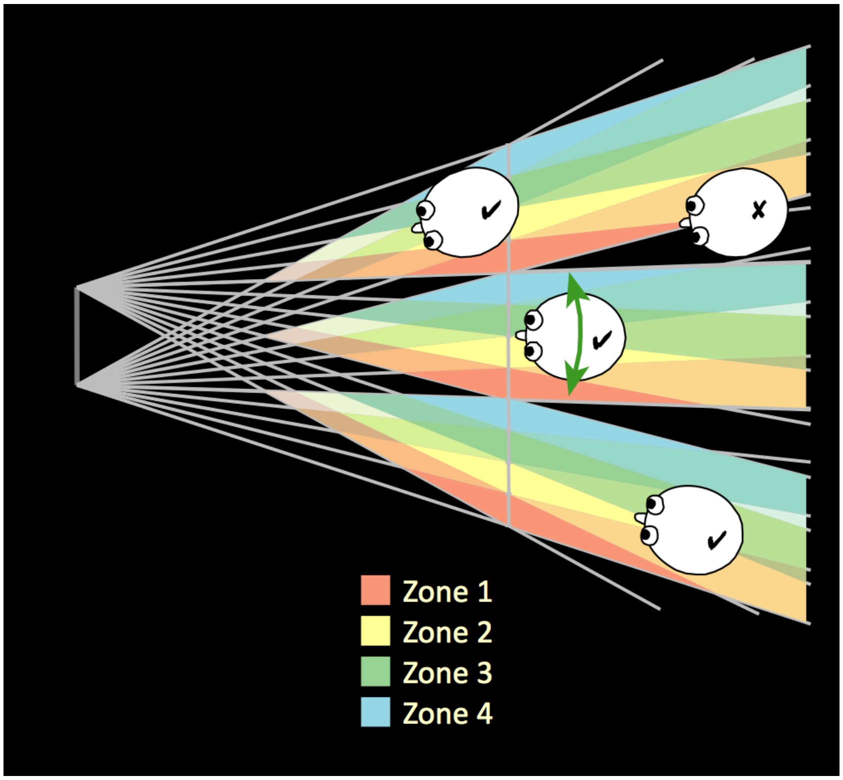
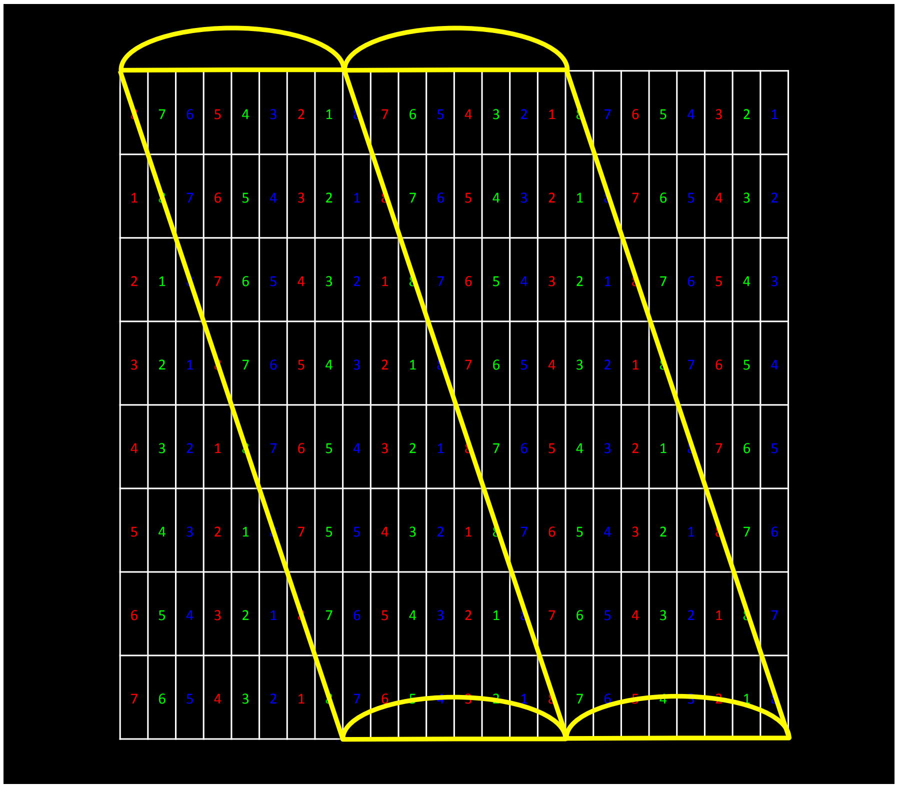


Figure 3.5: A 4-view lenticular display, permitting 4 distinct viewing zones [21].

zontal resolution of each of the views was shrunk by a factor of 4 to accommodate all 4 views. For a VGA display (640 × 480), the resolution of each of the underlying views would be terribly low at 160 × 480, making it unreasonable.

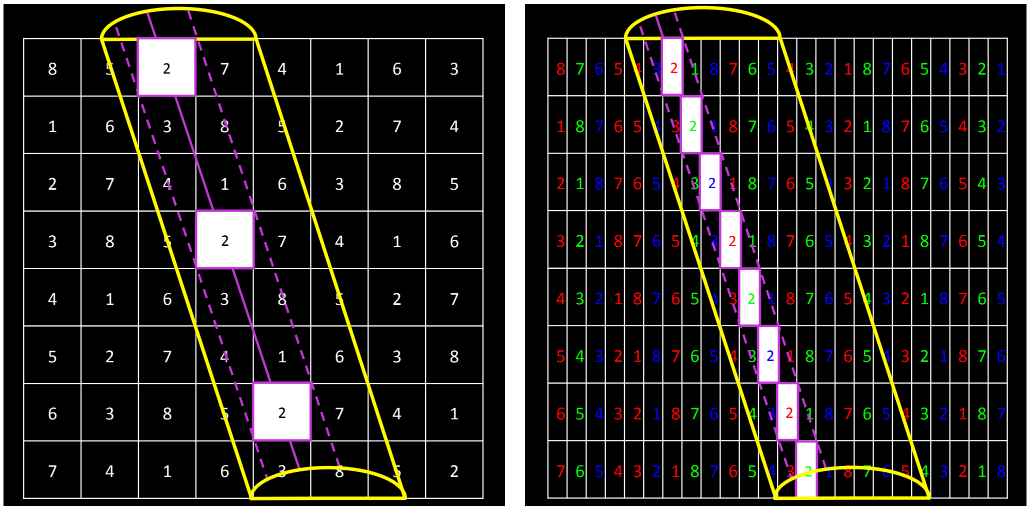
To mitigate the loss of horizontal resolution, Philips pioneered the slanted configuration for the lenticular array with their 7-view display [22]. By slanting the lenslets, they spread multiple views in the vertical direction as well the horizontal. The result was a reasonable resolution in both dimensions for all the seven views. Since then, a number of manufacturers have attempted to commercialize on similar designs, particularly Alioscopy, Inc. [23], which has focused on 8-view displays. Fig. 3.6 illustrates how these slanted lenslets are positioned for the Alioscopy display. In this figure, the number over each sub-pixel indicates the view to which the sub-pixel belongs and the color of the number indicates the respective color band (red, green, or blue) of that sub-pixel. By using this pattern, the individual views may be interspersed through the space in front of the display.



8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2
2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3
3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4
4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5
5	4	3	2	1	8	7	5	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6
6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7
7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1	8

Figure 3.6: Slanted lenticular lenslets, providing a larger number of viewing zones.

The innovative distinction over the original Philips design is that, by offsetting the three color channels vertically, Alioscopy managed to reduce the crosstalk between views in each of the viewing zones. To more clearly illustrate the point, Fig. 3.7a shows the band (purple region) on a specific lens that corresponds to the viewing zone for the second view without differentiating the sub-pixels. When viewed from this perspective at the optimal viewing distance, the observer's eye would ideally see information coming from pixels primarily in the second view. However, notice that in much of the band, information is also coming from views 1, 3, 4, and 8. The net result of this is that there is a diminished sense of depth perception and a general blurring of images. On the other hand, Fig 3.7b incorporates the offset sub-pixels. By diagonalizing the individual color components of each pixel, it effectively increases the LCD surface area corresponding to a viewing zone, significantly reducing information bleeding in from the other views. The result is a much more pleasant 3D experience with sharp edges.



(a) Pixel Zoning

(b) Offset Sub-Pixel Zoning

Figure 3.7: Highlighting a specific view zone. From this position on the lens, the second view would be most visible. Clearly using sub-pixels minimizes bleeding from other views into this zone.

Despite the many advances in fabricating high quality multiview autostereoscopic displays, manufacturers have not yet succeeded in getting them into the homes of consumers. The problem is that multiview content does not really exist, and perhaps never will in large quantities. On the other hand, single-view (2D) videos are ubiquitous. Even stereo content is growing at a significant rate. Therefore, the only options for generating content for multiview displays are to use computer graphics with 3D models or to build camera arrays with the necessary number of cameras.

It is impractical to use an array of eight or more cameras to collect multiview videos for most practices, and practically impossible in the realm of surgery. To provide a good 3D experience, the cameras would have to be nearly identical in their configurations (zoom, focus, color balance, synchronization, etc.), even down to their sensitivity to light at the sensor level. No two sensors are exactly the same, let alone 8 of them. Furthermore, it becomes a very complicated hardware problem. All cameras would have to be calibrated and placed in precise alignment. Any slight shift in physical position or optics or any lighting inconsistency would dramatically degrade the multiview effect. When considering surgical applications, having a large rig of multiple cameras makes no sense. To develop a camera system small enough for anatomical uses would require costly fabrication of a multi-lens sensor. In conclusion, without a software solution to convert stereoscopic data to multiview, autostereoscopic displays are by and large useless devices. Fortunately, we have developed a highly robust, real-time solution to deal with this problem and that will be the focus of the rest of this thesis.

Chapter 4

Disparity Estimation

4.1 Overview

Stereo depth estimation is an integral problem associated with the delivery of 3D content. Depth is an important element for an accurate visual representation of 3D content. Motivated by the human visual system, the problem is formulated as the determination of the distance of objects located in a scene based on stereo information. Humans see depth by integrating multiple visual cues and processing that information in their visual cortex. By far the most well studied cues come from having two eyes because they intrinsically correlate with depth. A simple experiment to validate the importance of binocular cues is to close one eye and try to grab something in front of you. Stereo depth estimation builds off this observation. Monocular cues (e.g., occlusion, motion, texture, relative size) may aid in the estimation, but stereo cues (e.g., horizontal parallax) are the primary and most robust indicators of depth.

In a two-camera imaging system, disparity is defined as the vector difference between the imaged object point in each image relative to the focal point [24]. It is this disparity that allows for depth estimation of objects in the scene via triangulation of the object point in each image. In rectified stereo, where both camera images are in the same plane, only horizontal disparity exists. In this case, multiview geometry shows that disparity is inversely proportional to the actual depth in the scene. Thus, if disparity can be measured from a rectified stereo

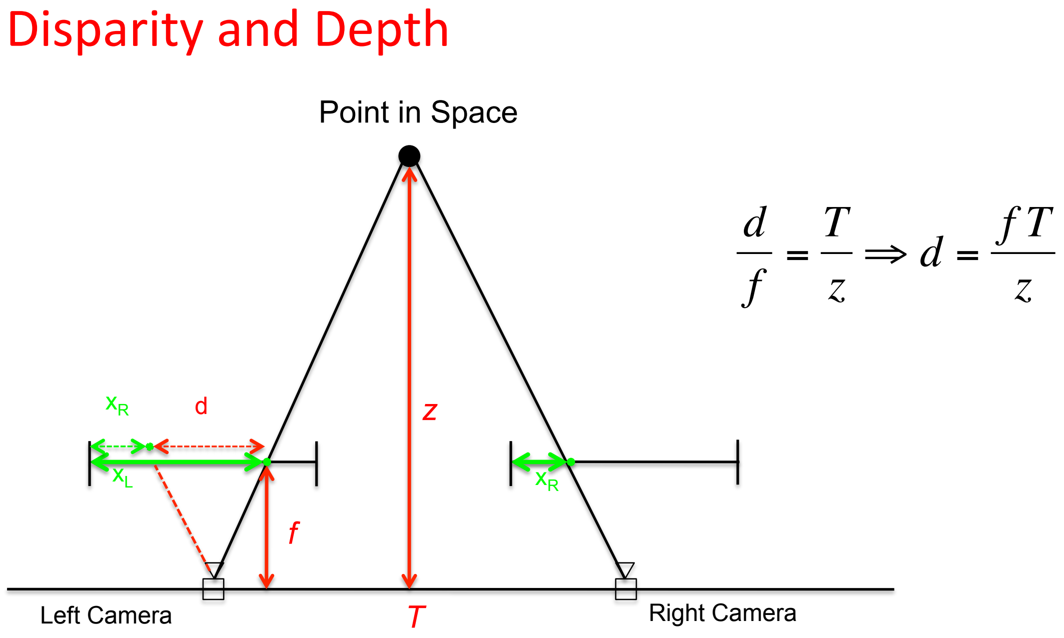


Figure 4.1: Relating depth to disparity.

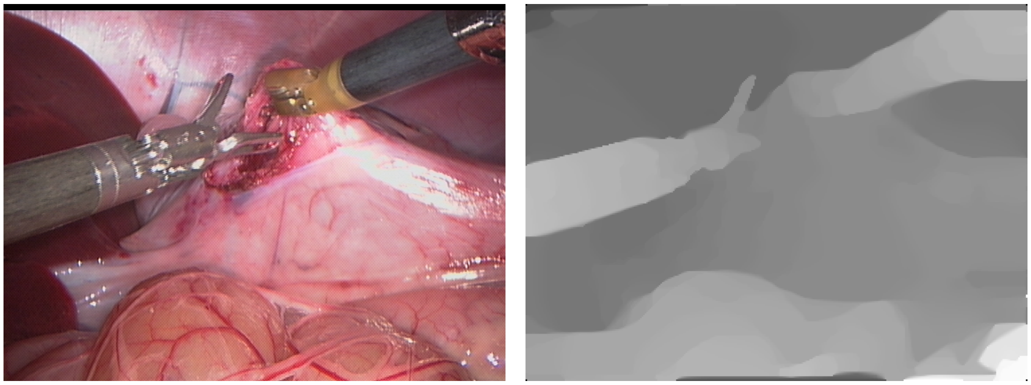
image pair, then the relative depth of each object can also be calculated. Fig. 4.1 shows the relationship between depth and disparity when the cameras are rectified and camera centers are known. The inverse relationship between depth z and disparity d is identified as:

$$d = \frac{fT}{z} \sim z^{-1} \tag{4.1}$$

where T is the camera separation (the interocular distance) and f is the camera focal length.

The problem of estimating disparity has been well-studied for images [25]. Excellent methods exist to estimate a disparity map, an image whose locations and intensities correspond to disparity magnitude at a given pixel location. Fig. 4.2 shows an example of an image and its associated disparity map.

The drawback with many of these existing algorithms is that they generally perform poorly on real-world images because they are trained on specific datasets, such as the Middlebury database [26]. Fig. 4.3 shows how the methods produce good estimates on the datasets for which they are created, but produce noisy, erroneous estimates when used on an arbitrary image.



(a) Left Image

(b) Left Disparity Image

Figure 4.2: An example of a dense disparity map generated from images taken by a stereo laparoscope. Here we show only the images for the left eye.

More importantly, applying these methods to a stereo video in a frame-by-frame basis is not guaranteed to produce spatially and temporally consistent disparity estimates. In particular, estimation error at object boundaries, occlusion regions, and textured areas may not be noticeable in one frame but can be apparent in a video, resulting in inconsistencies along the time axis. When these depth estimates are used for view synthesis in autostereoscopic displays, results are poor and contain a great deal of high-frequency flickering that significantly detracts from the 3D effect when the sequence is visualized.

In this chapter, we present a systematic approach by which we generate accurate and spatio-temporally consistent disparity maps from complex *stereo video sequences*, something that is absolutely crucial when working with surgical videos. We leverage the strengths of current state-of-the-art image-based techniques, but, in addition, we explicitly enforce the consistency of estimates in both space and time by treating the video as a space-time volume corrupted by noise. In doing so, we provide an algorithm that has the capability of refining arbitrary image-based disparity estimation techniques and, at the same time, extending them to the video domain.

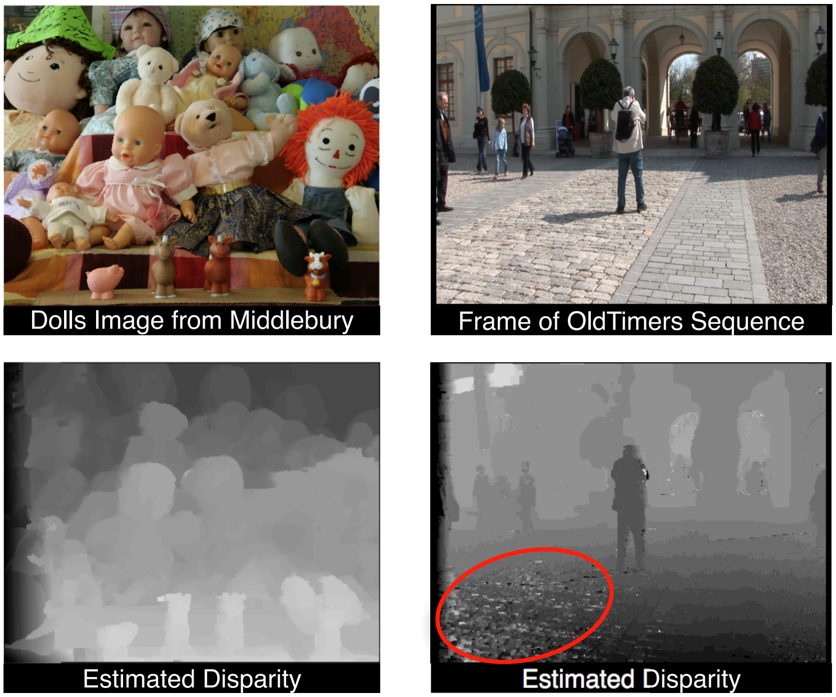


Figure 4.3: Lack of generalizability. Good estimates for Middlebury but noisy for OldTimers.

4.2 Related Work

4.2.1 Image Disparity Methods

For *static images*, the problem of disparity estimation has been thoroughly studied, with standardized databases, such as Middlebury [25], aiding in the fast evolution of the myriad techniques. The existing algorithms may be categorized into two groups: local and global methods. Local methods treat each pixel, or an aggregated region of pixels, in the reference image independently and try to infer the optimal horizontal displacement to match it with the corresponding pixel/region in the alternate image. In contrast, global methods incorporate assumptions about depth discontinuities and estimate disparity values by minimizing an energy function over all pixels using techniques like Graph Cuts [27, 28] or Hierarchical Belief Propagation (HBP) [29, 30]. Local methods tend to be fast but lack the accuracy of global methods. Yet, straightforward implementations of most global methods tend to be extremely slow. A thorough review of stereo matching techniques can be found in [31].

4.2.2 Video Disparity Methods

Stereo *video sequences*, on the other hand, have been studied less extensively than images. Solutions to the stereo matching problem are few and far between. Largely due to the computational bottleneck of dealing with multi-dimensional data, lack of any real datasets with ground-truth, and the unclear relationship between optimal spatial and temporal processing for correspondence matching, few have ventured to present viable solutions to the video disparity estimation problem. The ones that have tried, typically do so by directly extending existing methods for images to videos. Usually, such methods suffer from the debilitating computational complexity of having to minimize an energy function in a very large space, making them impractical for most applications.

In an attempt to build off the successful HBP approach, the authors in [32] considered a 3-dimensional Markov Random Field (a graph theoretic approach where pixels are treated as random variables with probabilistic interconnections) for HBP so that the temporal smoothness could be handled. This approach is slow, however, and computational times make it unusable in most cases. The reported algorithmic run-times are as high as 947.5 seconds (15 minutes) for a single 320×240 frame on a powerful computer.

Scene Flow [33] defined a 3D motion vector field and used it to improve temporal consistency. Similarly, [34] estimated motion using traditional optical flow and applied a median filter along the time axis as a post-processing step. Both methods required flow field estimation, however, which is computationally expensive if one desires high accuracy and which introduces unnecessary errors into the framework.

The method presented in [35] is one of the most promising techniques to the best of our knowledge, as it shows practical, real-time usability via a GPU implementation of HBP using an approximation to locally adaptive support weights [36]. The authors integrated temporal coherence in a similar way to [32] and also provided a synthetic dataset with ground-truth disparity maps. Even with their promising findings, however, our algorithm is capable of further refining their results.

Additional methods have also been proposed, but they typically require specific hardware, such as time-of-flight sensors [37], or constraints on the data, such as static scenes [38], that are beyond the scope of a surgical setting and so we do not consider them.

4.3 Approach

Our approach for disparity estimation [3, 1, 2] leverages the advances made with the image-based techniques. The core of the algorithm is fast, robust inferencing based on the method developed by Felzenszwalb et al., hierarchical belief propagation (HBP) [29], using locally adaptive support weights [36]. HBP has been shown to maintain the accuracy of global methods, such as traditional belief propagation or graph cuts, but rivals local methods in computational time. We now briefly review the method.

4.3.1 Review of Hierarchical Belief Propagation

HBP [29] begins by assuming a Markov Random Field (MRF) structure on the image plane. An MRF is an undirected graphical model where the random variables maintain the Markov Property (the probability distribution of future states only depends on the current state). In other words, each pixel in the image space is treated as a node (vertex) in a 4-connected grid (edges exist between vertical and horizontal neighbor pixels).

In this formulation, the framework of the problem becomes the following. Let $\mathcal{P}$ be the set of pixels in an image and $\mathcal{L}$ be a finite set of labels. The labels correspond to quantities that we want to estimate at each pixel (i.e., the disparity). A labeling f assigns a label $f_p \epsilon \mathcal{L}$ to each pixel $p \epsilon \mathcal{P}$. As with traditional global methods, for each pixel we designate an energy function that determines how well that label fits:

$$E(f) = \sum_{p \epsilon \mathcal{P}} D_p(f_p) + \sum_{(p,q) \epsilon \mathcal{N}} V(f_p - f_q) \tag{4.2}$$

$D_p(f_p)$ is referred to as the data cost and $V(f_p - f_q)$ is commonly called the smoothness cost in some literature, but it is more accurate to refer to it as a discontinuity cost. Intuitively, the data cost captures how well the labeling fits the node (how well the disparity estimate matches the stereo information). The discontinuity cost enforces the assumption that labels should vary slowly almost everywhere except for drastic changes along object boundaries. Neighboring pixels in the neighborhood $\mathcal{N}$ (the 4-connected grid) are penalized according to how large the difference is between their labels.

In our implementation, the data cost is computed over a large window for each pixel using Yoon and Kweon's locally adaptive support weights [36], so that only points with a high probability of belonging to the same object contribute significantly to the cost calculation. For the discontinuity cost, we use the commonly employed truncated weighted linear model,

$$V(f_p - f_q) = min(\alpha \left| f_p - f_q \right|, \beta), \tag{4.3}$$

where f_p and f_q are the labels we wish to assign to pixels p and q.

Minimizing the energy over the entire MRF is equivalent to computing the Maximum A Posteriori (MAP) estimate. Normally the MAP estimate would mean maximizing the product of the probability distributions, but the data and discontinuity costs can be seen as corresponding to negative log-likelihoods of the probabilities so that we are doing a minimization. The straightforward max-product BP algorithm produces an approximate solution to the MAP estimate by passing messages around the graph defined by the four-connected image grid. The method iteratively passes messages from all nodes in parallel. Each message is a vector of length equal to the number of possible labels (disparity levels). Using the notation of Felzenszwalb et al., let $m^t_{p \to q}$ be the message that node p sends to a neighboring node q at iteration t. At each iteration, new messages are computed in the following way:

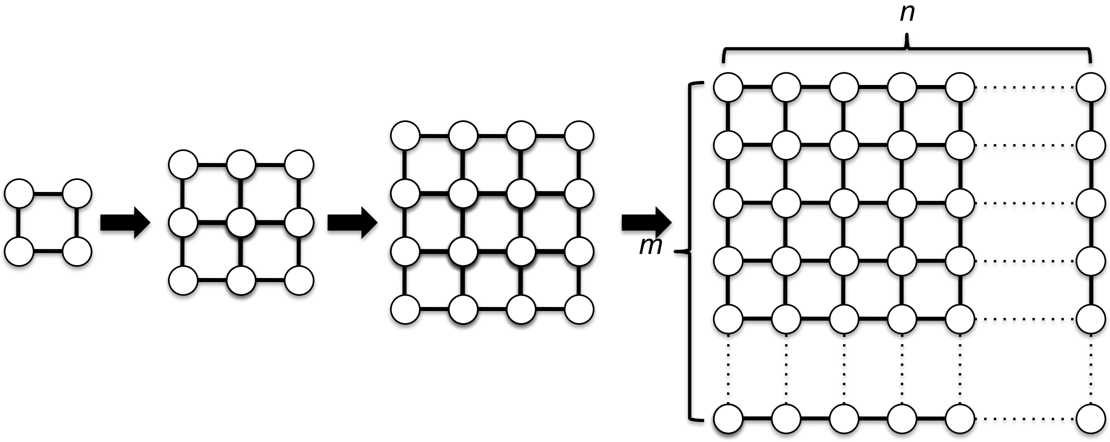


Figure 4.4: Belief Propagation is run hierarchically to dramatically speed up the standard algorithm. After each iteration, the results of the current level are passed down to the next lowest level until the full image resolution is reached.

$$\begin{aligned} m^t_{p\to q} &= \min_{f_p}\left(E(f) + \sum_{s\epsilon\mathcal{N}(p)\setminus q} m^{t-1}_{s\to p}(f_p)\right) \\ &= \min_{f_p}\left(D_p(f_p) + V(f_p - f_q) + \sum_{s\epsilon\mathcal{N}(p)\setminus q} m^{t-1}_{s\to p}(f_p)\right) \end{aligned} \tag{4.4}$$

Here $\mathcal{N}(p) \setminus q$ denotes neighbors of p other than q. The message vector represents the minimal-energy labeling of node p and all the information coming into it through the connected nodes. The idea is that after T iterations, information from one side of the image will have propagated to the other side. Then a belief vector is generated as:

$$b_q(f_q) = D_q(f_q) + \sum_{p\epsilon\mathcal{N}(q))} m^T_{p\to q}(f_q) \tag{4.5}$$

The final labeling is then selected as the label that minimizes the belief vector at each node individually. After sufficient iterations, the minimization will lead to a globally optimal disparity labeling across the entire image.

A naive implementation of this algorithm is computationally very slow, but Felzenszwalb et al. show that a number of techniques can be used to significantly reduce computational time. Most notably, by computing the message vectors on multiple resolutions, there can be a significant speedup. With this technique, Belief Propagation is run on a coarse-to-fine manner as shown in Fig. 4.4. At each level in the hierarchy, messages are computed and then passed down to the lower level until the full-resolution level is reached and the final inference is made. This approach allows messages to propagate throughout the entire MRF (Markov Random Field) but with much fewer iterations. As a result, we are capable of generating robust disparity estimates at a fraction of the computational time of normal BP or graph cuts algorithms.

Chapters 4, 5, and 8, in part, are reprints of the material as it appears in [1, 2, 3]. The dissertation author was the primary investigator and author in [1] and secondary in [2, 3].

Chapter 5

Spatio-temporally Consistent Disparity Estimation

5.1 Introduction

In the previous chapter, we discussed how disparity maps may be computed individually on a frame-by-frame basis. The difficulty with operating on each frame of a video sequence independently is that the consistency between consecutive frames is lost. The noisy estimates for each frame create a flickering effect over time that is highly bothersome to the HVS. To compensate for the temporal inconsistency, we present a novel, fast, and efficient method.

The key advantage is that we avoid reformulating the problem in space-time, as a number of others have tried, because of the realization that such attempts become computationally impractical. Furthermore, image-based techniques have been thoroughly studied and are much more advanced in their implementations so we wish to leverage all the breakthroughs made with them.

The proposed method is a two-stage algorithm, as depicted in Fig. 5.1. In the first stage, our image-based method (or any other existing disparity estimation method) is applied to individual frames of the video to generate initial estimates. Then between neighboring pixels in space-time, we enforce the disparity smoothness assumption by mandating that values should vary smoothly except at object

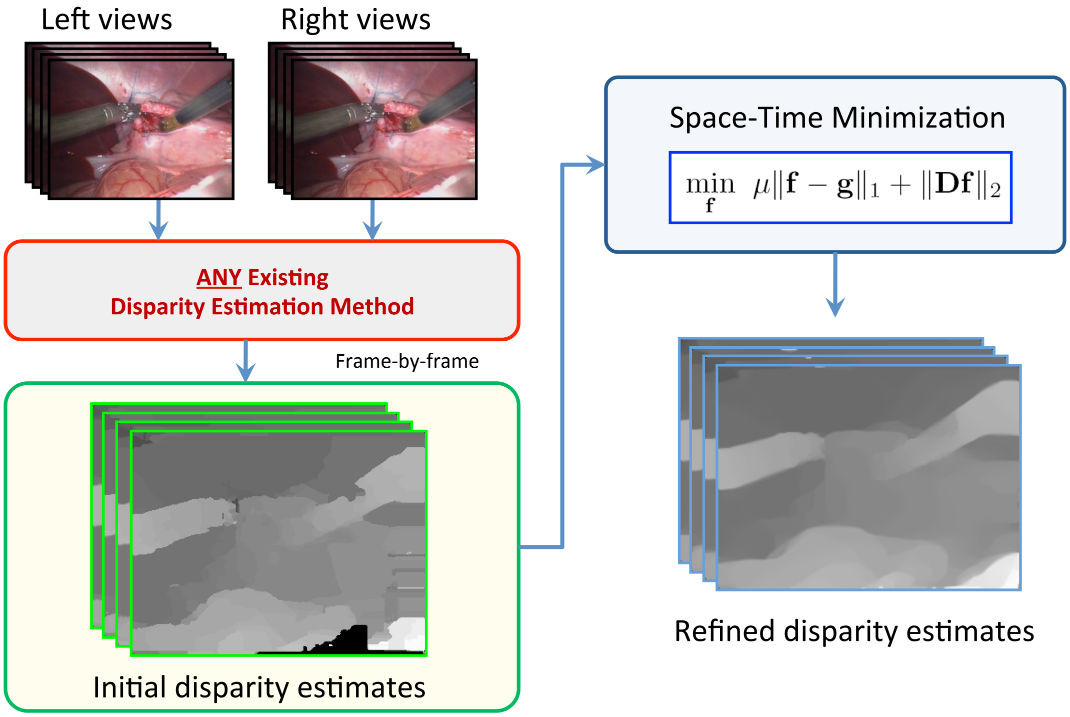

Left views
Right views
ANY Existing
Disparity Estimation Method
Frame-by-frame
Initial disparity estimates
Space-Time Minimization
$\min_{\mathbf{f}} \ \mu\|\mathbf{f}-\mathbf{g}\|_1 + \|\mathbf{Df}\|_2$
Refined disparity estimates

Figure 5.1: Block diagram of the proposed method. In the first step, *any* existing disparity estimation method can be used to generate initial disparity maps. In the second step, a space-time minimization problem is solved to enhance the spatial and temporal consistency of the initial estimates.

boundaries. This is because objects do not simply appear and disappear from one frame to the next. However, this smoothness assumption is normally violated in most initial disparity estimates, as there are inevitable estimation errors. Thus in the second stage, the initial estimates are formed into a three-dimensional volume in space-time, denoted as the space-time volume. Then a space-time minimization problem is solved to enforce spatial and temporal consistency in this volume.

Although our emphasis is on videos, it is important to note that image disparity can also be improved using the proposed algorithm because images can be treated as videos with just a single frame. Later in this chapter, we will show that the proposed algorithm reduces estimation error for ***all*** *top-ranking image-based disparity estimation techniques on the Middlebury evaluation website* [26].

5.2 Background on Video Restoration

Image restoration is an inverse problem where the objective is to recover a sharp image from a blurry and noisy observation. Mathematically, a linear shift invariant imaging system is modelled as [39]

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \eta \tag{5.1}$$

where $\mathbf{f} \in \mathbb{R}^{MN\times 1}$ is a vector denoting the unknown (potentially sharp) image of size $M \times N$, $\mathbf{g} \in \mathbb{R}^{MN\times 1}$ is a vector denoting the observed image, $\eta \in \mathbb{R}^{MN\times 1}$ is a vector denoting the noise, and the matrix $\mathbf{H} \in \mathbb{R}^{MN\times MN}$ is a linear transformation representing the convolution operation. The goal of image restoration is to recover $\mathbf{f}$ from the observed image $\mathbf{g}$.

Standard single image restoration has been studied for more than half a century. Popular methods such as Wiener deconvolution [39], Lucy Richardson deconvolution [40, 41], and regularized least squares minimization [42, 43] have already been integrated into MATLAB. Advanced methods such as total variation image restoration methods are also becoming mature [44, 45, 46, 47, 48].

While single image restorations still have room for improvement, we must consider the video restoration problem. The key difference between images and videos is the additional time dimension. Consequently, video restoration has some unique features that do not exist in image restoration.

1. **Motion Information**
   Motion deblurring requires a motion vector field, which can be estimated from a video sequence using conventional methods, such as block matching [49] and optical flow [50]. While it is also possible to remove motion blur based on a single image, for example [51, 52, 53, 54, 54], the performance is limited to global motion or at most one or two objects by using sophisticated object segmentation algorithms.

2. **Spatial Variance versus Spatial Invariance**
   For a class of spatially variant image restoration problems (in particular, motion blur), the convolution matrix $\mathbf{H}$ is not a block-circulant matrix.

Therefore, Fourier Transforms cannot be utilized to efficiently find a solution. Videos, in contrast, allow us to transform a sequence of spatially variant problems to a spatially *invariant* problem.

3. **Temporal Consistency**

   Temporal consistency is concerned with the smoothness of the restored video along the time axis. Although smoothing can be performed spatially (as in the case of single image restoration), temporal consistency cannot be guaranteed if these methods are applied to a video in a frame-by-frame basis.

   Because of these unique features in video, we may seek a video restoration algorithm that utilizes motion information, exploits the spatially invariant properties, and enforces spatial and temporal consistency.

Most of the current state-of-the-art video restoration methods recover a video by solving a sequence of individual image restorations. To improve the temporal consistency among the frames, various approaches have been adopted: [55] modified Eq. (5.1) to incorporate the geometric warping caused by motion; [56] utilized the motion vector field as a prior to the restoration; and [57] considered a regularization function of the residue between the current solution and the motion compensated version of the previous solution.

Another class of methods are based on the concept of the "space-time volume," which was first introduced in the early 1990s by [58], and rediscovered by Wexler, Shechtman, Caspi, and Irani [59, 60]. The idea of the space-time volume is to stack the frames of a video to form a three-dimensional data structure known as the space-time volume. This allows one to transform the spatially variant motion blur problem into a spatially invariant one. By imposing regularization functions along both the spatial and temporal directions, both spatial and temporal smoothness can be enforced. However, the size of the space-time volume is much larger than a single image. Therefore, the authors of [60] only considered a regularized least-squares minimization, as there is a closed-form solution. More sophisticated regularization functions, such as total variation [55] and its l_1-approximation [61] did not seem possible under this framework, for these non-differentiable functions are difficult to solve efficiently.

Our work is based on the augmented Lagrangian method, an old method that has recently drawn significant attention [47, 48, 62]. All of these methods follow from Eckstein and Bertsekas' operator splitting method [63], which can be traced back to the work of Douglas and Rachford [64] and the proximal point algorithm [65, 66]. There has been no work, however, on extending the augmented Lagrangian method to space-time minimization. Before we discuss our approach, we begin by discussing why the space-time volume can be applicable for video disparity estimation in the first place.

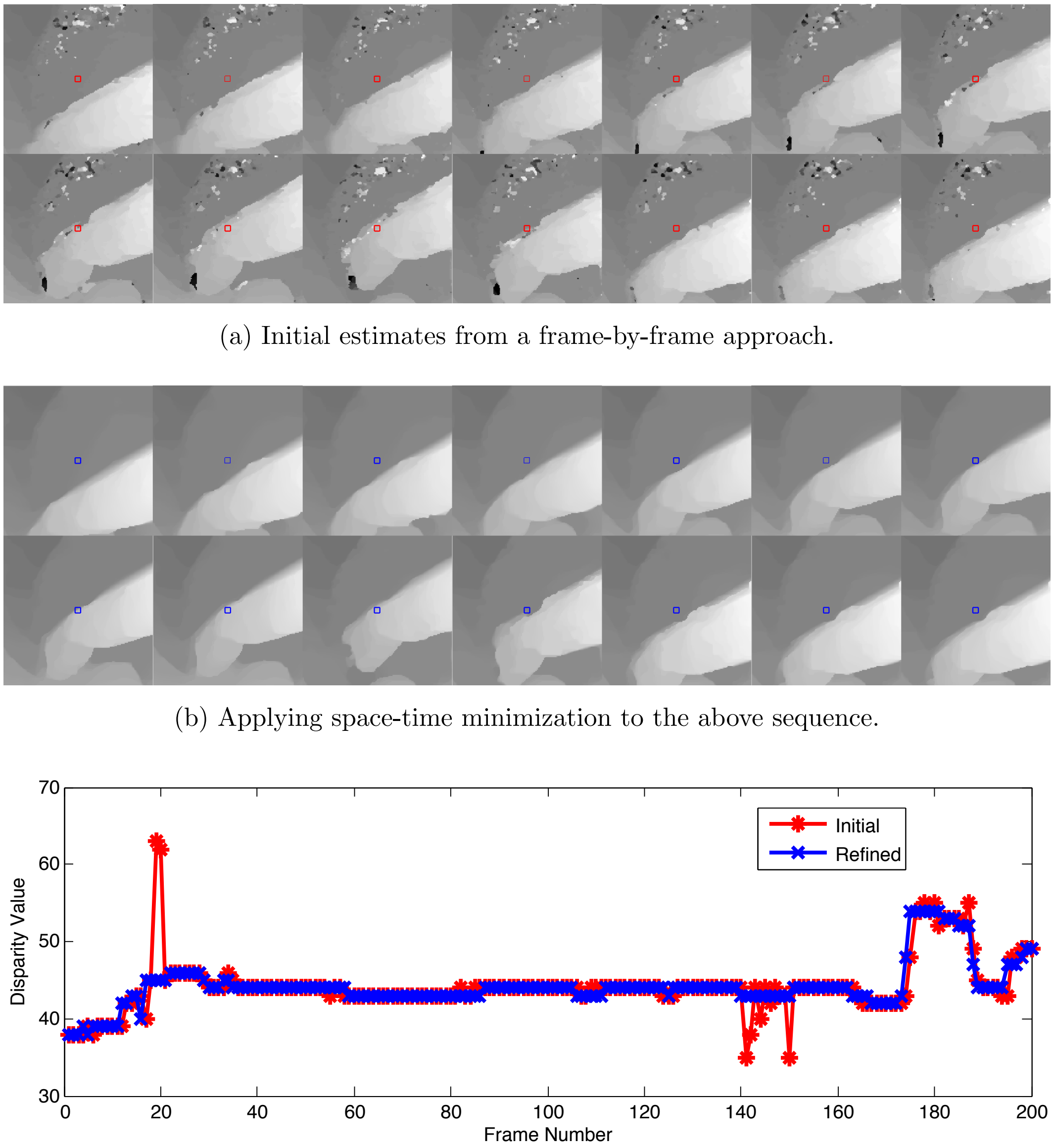
5.3 Discussing the Space-Time Volume

We begin by inspecting the problem of using image-based algorithms to generate video disparity estimates. The algorithm presented here is the one we presented in the previous chapter, but other algorithms can be similarly analysed.

Fig. 5.2 shows a few snapshots of video disparity estimates using our image-based implementation. At any fixed instant in time, it can be observed that the disparity is noisy, especially at object boundaries. The cause of these noisy disparity estimates varies from algorithm to algorithm, but inefficient occlusion handling and lighting imbalances are two key contributors. We refer to this type of disparity instability as *spatial* inconsistency since it is independent of time.

For a fixed pixel location marked by a red box in Fig. 5.2, a red-colored curve showing the disparity label as a function of time is plotted in the Fig. 5.2(c). The fluctuation of the disparity in time is referred to as *temporal* inconsistency, which is caused by the inability of the algorithm to handle temporal smoothness. Note that temporal consistency is closely related to spatial consistency, because a temporally noisy disparity is also likely to be spatially noisy (and vice versa).

Our approach to improving the spatio-temporal consistency of a disparity sequence is to treat noisy estimates as *outliers.* That is, we seek a disparity which tries its best to fit the initial estimate, while at the same time eliminating outliers. To this end, an l_1-norm regression approach is considered as it is more robust than



(a) Initial estimates from a frame-by-frame approach.

(b) Applying space-time minimization to the above sequence.

(c) Normalized disparity as a function of time. The red and blue curves correspond to the mean disparity in the red and blue boxes in (a) and (b).

Figure 5.2: Snapshots of estimated disparity. Note that the result is spatially and temporally more consistent after space-time minimization.

an l_2-norm regression. Since l_1-minimization is a convex problem, designing an efficient algorithm to find the global minimum is possible.

5.4 Proposed Minimization Algorithm

Let $f(x, y, t)$ be the disparity at (x, y) in space and t in time (See Fig. 5.1). For notational simplicity we stack $f(x, y, t)$ to form a column vector $\mathbf{f}$ (i.e., $\mathbf{f} = \mathbf{vec}(f(x, y, t))$). If $f(x, y, t)$ has M rows, N columns and K frames, then $\mathbf{f}$ is a vector of dimension $MNK \times 1$. The i-th element of $\mathbf{f}$ is denoted by $[\mathbf{f}]_i$.

Given a sequence of initial disparity maps $\mathbf{g}$, our goal is to reduce the effects of the outliers while fitting the initial estimates. Mathematically, we seek to solve a minimization problem

$$\underset{\mathbf{f}}{\text{argmin}} \quad \mu\Psi(\mathbf{f}) + \Phi(\mathbf{f}), \tag{5.2}$$

where μ is a regularization parameter, $\Psi(\mathbf{f})$ encapsulates the fidelity residue (Section 5.4.1), and $\Phi(\mathbf{f})$ represents a regularization that controls the smoothness (Section 5.4.2).

5.4.1 Fidelity

We define the objective function in Eq. (5.2) as the l_1-norm of the residue $\mathbf{f} - \mathbf{g}$:

$$\Psi(\mathbf{f}) = \|\mathbf{f} - \mathbf{g}\|_1. \tag{5.3}$$

The motivation, as mentioned in Section 5.3, is that disparity estimation errors can be considered as outliers, and an l_1-denoising algorithm is effective at eliminating noise while preserving sharp edges.

In fact, $\|\mathbf{f} - \mathbf{g}\|_1$ is related to the notion of the percentage of bad pixels, a quantity commonly used to evaluate disparity estimation algorithms. Given a ground truth disparity $\mathbf{f}^*$, the number of bad pixels of an estimated disparity $\mathbf{f}$ is the cardinality of the set

$$\{i| \ \ |[\mathbf{f} - \mathbf{f}^*]_i| > \tau\} \tag{5.4}$$

for some threshold τ. In the absence of ground truth, the same idea can be used with a reference disparity (*e.g.*, $\mathbf{g}$). In this case, the cardinality of the set

$$\Omega_\tau = \{i| \ \ |[\mathbf{f} - \mathbf{g}]_i| > \tau\}, \tag{5.5}$$

denoted by $|\Omega_\tau|$, is the number of bad pixels of $\mathbf{f}$ with respect to (w.r.t) $\mathbf{g}$. Therefore, minimizing $|\Omega_\tau|$ is equivalent to minimizing the number of bad pixels of $\mathbf{f}$ w.r.t. $\mathbf{g}$. However, this problem is non-convex and NP-hard. In order to alleviate the computational difficulty, we set $\tau = 0$ so that $|\Omega_\tau| = \|\mathbf{f} - \mathbf{g}\|_0$, and convexify (make into a convex function) $\|\mathbf{f} - \mathbf{g}\|_0$ by $\|\mathbf{f} - \mathbf{g}\|_1$. Therefore, $\|\mathbf{f} - \mathbf{g}\|_1$ does not only represent a regression, but can also be interpreted as a convexification of the notion of the percentage of bad pixels.

5.4.2 Smoothness in Space-Time

To simultaneously enforce smoothness in space and time while maintaining fidelity, we propose to consider the space-time total variation (TV) regularization. We define three forward-difference operators (matrices) $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$ as

$$\begin{aligned}
\mathbf{D}_x\mathbf{f} &= \mathbf{vec}(f(x+1,y,t) - f(x,y,t)) \\
\mathbf{D}_y\mathbf{f} &= \mathbf{vec}(f(x,y+1,t) - f(x,y,t)) \\
\mathbf{D}_t\mathbf{f} &= \mathbf{vec}(f(x,y,t+1) - f(x,y,t)).
\end{aligned}$$

Note that the operation of $\mathbf{D}_x$ on $\mathbf{f}$ can be performed using a convolution with the kernel $[1, \ -1]$. Therefore, the matrix $\mathbf{D}_x$ can be shown to be block-circulant. This property also applies to $\mathbf{D}_y$ and $\mathbf{D}_t$.

In order to have greater flexibility in controlling the regularization terms, we introduce three scaling factors to the forward difference operators as follows. We define the scalars β_x, β_y, and β_t and multiply them with $\mathbf{D}_x$, $\mathbf{D}_y$, and $\mathbf{D}_t$ respectively so that

$$\mathbf{D} = [\beta_x \mathbf{D}_x^T, \ \beta_y \mathbf{D}_y^T, \ \beta_t \mathbf{D}_t^T]^T \tag{5.6}$$

As a result, with $(\beta_x, \beta_y, \beta_t)$, the space-time total variation norm on $\mathbf{f}$ can be expressed as

$$\|\mathbf{D}\mathbf{f}\|_2 = \sum_i \sqrt{\beta_x^2 \left[\mathbf{D}_x\mathbf{f}\right]_i^2 + \beta_y^2 \left[\mathbf{D}_y\mathbf{f}\right]_i^2 + \beta_t^2 \left[\mathbf{D}_t\mathbf{f}\right]_i^2}. \tag{5.7}$$

Eq. (5.7) is a generalization of conventional total variation. If $\beta_x = \beta_y = 1$, and $\beta_t = 0$, then

$$\|\mathbf{D}\mathbf{f}\|_2 = \sum_i \sqrt{[\mathbf{D}_x\mathbf{f}]_i^2 + [\mathbf{D}_y\mathbf{f}]_i^2} \tag{5.8}$$

is the two-dimensional total variation in the spatial domain. If $\beta_x = \beta_y = 0$ and $\beta_t = 1$, then $\|\mathbf{Df}\|_2 = \|\mathbf{D}_t\mathbf{f}\|_1$ becomes the one-dimensional total variation in the temporal domain. By adjusting the relative weights $(\beta_x, \beta_y, \beta_t)$, we can control the relative emphasis put on the individual terms $\mathbf{D}_x\mathbf{f}$, $\mathbf{D}_y\mathbf{f}$ and $\mathbf{D}_t\mathbf{f}$. In short, we define our regularization term, $\Phi(\mathbf{f})$, as:

$$\Phi(\mathbf{f}) = \|\mathbf{Df}\|_2. \tag{5.9}$$

5.4.3 Solving the Minimization Problem

Combining Eqs. (5.3) and (5.9) yields the optimization problem that we wish to solve

$$\underset{\mathbf{f}}{\operatorname{argmin}} \quad \mu\|\mathbf{f} - \mathbf{g}\|_1 + \|\mathbf{Df}\|_2. \tag{5.10}$$

Problem (5.10) is known as the TV/L1 minimization problem [48] and can be solved using the Douglas-Rachford operator splitting method [63]. Our implementation follows from [67], which has better convergence properties than [48]. However, some modifications must be made to accommodate for the three-dimensional data structure of a space-time signal.

Following the idea of the Douglas-Rachford operator splitting method, we introduce two intermediate variables $\mathbf{u}$ and $\mathbf{r}$ so that (5.10) can be transformed into an equivalent constrained problem

$$\begin{aligned} \underset{\mathbf{f},\mathbf{u},\mathbf{r}}{\operatorname{argmin}} \quad & \mu\|\mathbf{r}\|_1 + \|\mathbf{u}\|_2 \\ \text{subject to} \quad & \mathbf{r} = \mathbf{f} - \mathbf{g} \;\text{ and }\; \mathbf{u} = \mathbf{Df}, \end{aligned} \tag{5.11}$$

where $\|\mathbf{u}\|_2 = \sum_i \sqrt{[\mathbf{u}_x]_i^2 + [\mathbf{u}_y]_i^2 + [\mathbf{u}_t]_i^2}$ and $\mathbf{u} = [\mathbf{u}_x^T,\, \mathbf{u}_y^T,\, \mathbf{u}_t^T]^T$. The constrained problem (5.11) can be solved by finding a saddle point of the augmented Lagrangian function

$$\begin{aligned} L(\mathbf{f}, \mathbf{r}, \mathbf{u}, \mathbf{y}, \mathbf{z}) = \mu \left\|\mathbf{r}\right\|_1 + \left\|\mathbf{u}\right\|_2 & \\ - \mathbf{z}^T(\mathbf{r} - \mathbf{f} + \mathbf{g}) + \frac{\rho_o}{2} \left\|\mathbf{r} - \mathbf{f} + \mathbf{g}\right\|_2^2 & \\ - \mathbf{y}^T(\mathbf{u} - \mathbf{Df}) + \frac{\rho_r}{2} \left\|\mathbf{u} - \mathbf{Df}\right\|_2^2, & \end{aligned} \tag{5.12}$$

where $\mathbf{y}$ is the Lagrange multiplier associated with constraint $\mathbf{u} = \mathbf{Df}$ and $\mathbf{z}$ is the Lagrange multiplier associated with the constraint $\mathbf{r} = \mathbf{f} - \mathbf{g}$. Here ρ_o and ρ_r are two regularization parameters associated with the quadratic penalty terms $\|\mathbf{r} - \mathbf{f} + \mathbf{g}\|_2^2$ and $\|\mathbf{u} - \mathbf{Df}\|_2^2$ respectively.

The idea behind the augmented Lagrangian method is to find a saddle point of the augmented Lagrangian function $L(\mathbf{f}, \mathbf{r}, \mathbf{u}, \mathbf{y}, \mathbf{z})$. To this end, we use the alternating direction method (ADM). At the k-th iteration, the alternating direction method solves the following sub-problems:

$$\begin{aligned}
\mathbf{f}_{k+1} &= \underset{\mathbf{f}}{\operatorname{argmin}} \quad L(\mathbf{f}, \mathbf{r}_k, \mathbf{u}_k, \mathbf{y}_k, \mathbf{z}_k) && (5.13)\\
&= \underset{\mathbf{f}}{\operatorname{argmin}} \quad \mathbf{z}_k^T \mathbf{f} + \frac{\rho_o}{2} \|\mathbf{r}_k - \mathbf{f} + \mathbf{g}\|_2^2 + \mathbf{y}_k^T \mathbf{Df} + \frac{\rho_r}{2} \|\mathbf{u}_k - \mathbf{Df}\|_2^2,\\
\mathbf{r}_{k+1} &= \underset{\mathbf{r}}{\operatorname{argmin}} \quad L(\mathbf{f}_{k+1}, \mathbf{r}, \mathbf{u}_k, \mathbf{y}_k, \mathbf{z}_k) && (5.14)\\
&= \underset{\mathbf{r}}{\operatorname{argmin}} \quad \mu \|\mathbf{r}\|_1 - \mathbf{z}_k^T(\mathbf{r} - \mathbf{f}_{k+1} + \mathbf{g}) + \frac{\rho_o}{2} \|\mathbf{r} - \mathbf{f}_{k+1} + \mathbf{g}\|_2^2,\\
\mathbf{u}_{k+1} &= \underset{\mathbf{u}}{\operatorname{argmin}} \quad L(\mathbf{f}_{k+1}, \mathbf{r}_{k+1}, \mathbf{u}, \mathbf{y}_k, \mathbf{z}_k) && (5.15)\\
&= \underset{\mathbf{u}}{\operatorname{argmin}} \quad \|\mathbf{u}\|_2 - \mathbf{y}_k^T(\mathbf{u} - \mathbf{Df}_{k+1}) + \frac{\rho_r}{2} \|\mathbf{u} - \mathbf{Df}_{k+1}\|_2^2.
\end{aligned}$$

It also updates the Lagrange multipliers as:

$$\begin{aligned}
\mathbf{y}_{k+1} &= \mathbf{y}_k - \rho_r(\mathbf{u}_{k+1} - \mathbf{Df}_{k+1}),\\
\mathbf{z}_{k+1} &= \mathbf{z}_k - \rho_o(\mathbf{r}_{k+1} - (\mathbf{f}_{k+1} - \mathbf{g})).
\end{aligned}$$

The ‘f’ sub-problem

Sub-problem (5.13) is differentiable and so by the first-order optimality criteria, we derive the normal equations

$$(\rho_o + \rho_r \mathbf{D}^T\mathbf{D})\mathbf{f} = \rho_o \mathbf{g} + (\rho_o \mathbf{r}_k - \mathbf{z}_k) + \mathbf{D}^T(\rho_r \mathbf{u}_k - \mathbf{y}_k). \qquad (5.16)$$

Since the operators $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$ are block-circulant matrices, it can be shown that the matrix $\rho_o + \rho_r \mathbf{D}^T\mathbf{D}$ is diagonalizable using a three-dimensional (3D) Fourier Transform:

$$\mathbf{F}(\rho_o + \rho_r \mathbf{D}^T\mathbf{D})\mathbf{F}^H = \rho_o + \rho_r(\beta_x^2|\mathbf{\Lambda}_{\mathbf{D}_x}|^2 + \beta_y^2|\mathbf{\Lambda}_{\mathbf{D}_y}|^2 + \beta_t^2|\mathbf{\Lambda}_{\mathbf{D}_t}|^2), \qquad (5.17)$$

where $\mathbf{F}$ is the discrete 3D Fourier Transform matrix, $(\cdot)^H$ is the Hermitian operator and $\mathbf{\Lambda}_{\mathbf{D}_x}$, $\mathbf{\Lambda}_{\mathbf{D}_y}$ and $\mathbf{\Lambda}_{\mathbf{D}_t}$ are the eigenvalue matrices of $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_t$ respectively. With the diagonalization (5.17), the normal equations (5.16) can be solved in three steps to determine $\mathbf{f}$:

(i) Apply the discrete 3D Fourier Transform to $\rho_o\mathbf{g}+(\rho_o\mathbf{r}_k-\mathbf{z}_k)+\mathbf{D}^T(\rho_r\mathbf{u}_k-\mathbf{y}_k)$.

(ii) Perform element-wise division by $(\rho_o+\rho_r(\beta_x^2|\mathbf{\Lambda}_{\mathbf{D}_x}|^2+\beta_y^2|\mathbf{\Lambda}_{\mathbf{D}_y}|^2+\beta_t^2|\mathbf{\Lambda}_{\mathbf{D}_t}|^2))$.

(iii) Apply an inverse discrete 3D Fourier Transform.

The 'r' sub-problem

Sub-problem (5.14) is given by

$$\underset{\mathbf{r}}{\text{argmin}} \quad \mu\left\|\mathbf{r}\right\|_1-\mathbf{z}^T\mathbf{r}+\frac{\rho_o}{2}\left\|\mathbf{r}-\mathbf{f}+\mathbf{g}\right\|_2^2,$$

which is known as the shrinkage problem. As shown in [67], the shrinkage problem has a closed form solution

$$\mathbf{r}_{k+1}=\max\left\{|\mathbf{w}|-\frac{\mu}{\rho_o},0\right\}\cdot\text{sign}\left(\mathbf{w}\right),$$

where $\cdot$ denotes component-wise multiplication, $\mathbf{w}=\mathbf{f}_{k+1}-\mathbf{g}+\frac{1}{\rho_o}\mathbf{z}_k$, and $|\mathbf{w}|$ is the component-wise modulus of $\mathbf{w}$.

The 'u' sub-problem

Similar to the *'r' sub-problem*, the solution to (5.15) can be found using the shrinkage formula as well

$$\mathbf{u}_{k+1}=\max\left\{\|\mathbf{v}\|_2-\frac{1}{\rho_r},0\right\}\cdot\frac{\mathbf{v}}{\|\mathbf{v}\|_2},$$

where $\mathbf{v}=[\mathbf{v}_x^T,\ \mathbf{v}_y^T,\ \mathbf{v}_t^T]^T$, with components

$$\begin{aligned}\mathbf{v}_x&=\beta_x\mathbf{D}_x\mathbf{f}+\tfrac{1}{\rho_r}\mathbf{y}_x\\ \mathbf{v}_y&=\beta_y\mathbf{D}_y\mathbf{f}+\tfrac{1}{\rho_r}\mathbf{y}_y\\ \mathbf{v}_t&=\beta_t\mathbf{D}_t\mathbf{f}+\tfrac{1}{\rho_r}\mathbf{y}_t\\ \|\mathbf{v}\|_2&=\textstyle\sum_i\sqrt{[\mathbf{v}_x]_i^2+[\mathbf{v}_y]_i^2+[\mathbf{v}_t]_i^2},\end{aligned}$$

Updating the regularization parameters

The Lagrange multipliers, $\mathbf{y}$ and $\mathbf{z}$ are partitioned as $\mathbf{y} = [\mathbf{y}_x^T,\ \mathbf{y}_y^T,\ \mathbf{y}_t^T]^T$ and $\mathbf{z} = [\mathbf{z}_x^T,\ \mathbf{z}_y^T,\ \mathbf{z}_t^T]^T$ respectively. The corresponding parameters, ρ_r and ρ_o, are updated as

$$\rho_r = \begin{cases} 2\rho_r, & \text{if } \|\mathbf{u}_{k+1} - \mathbf{D}\mathbf{f}_{k+1}\|_2 \geq \alpha_r \|\mathbf{u}_k - \mathbf{D}\mathbf{f}_k\|_2, \\ \rho_r, & \text{otherwise.} \end{cases}$$

$$\rho_o = \begin{cases} 2\rho_o, & \text{if } \|\mathbf{g} - \mathbf{f}_{k+1}\|_2 \geq \alpha_o \|\mathbf{g} - \mathbf{f}_k\|_2, \\ \rho_o, & \text{otherwise.} \end{cases}$$

where $\alpha_r = \alpha_o = 0.7$. The intuition is that each of the quadratic penalties, $\frac{\rho_r}{2}\|\mathbf{u} - \mathbf{D}\mathbf{f}\|_2^2$ and $\frac{\rho_o}{2}\|\mathbf{r} - \mathbf{f} + \mathbf{g}\|_2^2$, is a convex surface added to the original objective function so that the problem is guaranteed to be strongly convex [65]. Ideally, the residues, $\frac{\rho_r}{2}\|\mathbf{u}_k - \mathbf{D}\mathbf{f}_k\|_2^2$ and $\frac{\rho_o}{2}\|\mathbf{r}_k - \mathbf{f}_k + \mathbf{g}\|_2^2$, should decrease as k increases. However, if they are not decreasing, the weight of the penalty must be increased relative to the objective function so that they will be forced to be reduced. Therefore, the parameter update scheme makes sure that the constraint violation is decreasing asymptotically. In the steady state as $k \to \inf$, the parameters will become constants [68].

The parameter μ balances a trade-off between the error in fidelity and the total variation penalty. Large values of μ tend to give sharper results, but the noise will be amplified. Small values of μ give less noisy results, but the video frames may become too smooth. Empirically, the optimal value for μ falls in the range of $0.01 \leq \mu \leq 3$ for disparity estimation. A default value of $\mu = 1$ works fairly well in general. To find the optimal value, we may start the algorithm with a small μ and exhaustively search for the best one. If the best μ hits either the lower or upper bounds, then the algorithm returns the initial estimate as its output, indicating that the initial disparity is already spatio-temporally smooth.

Chapters 4, 5, and 8, in part, are reprints of the material as it appears in [1, 2, 3]. The dissertation author was the primary investigator and author in [1] and secondary in [2, 3].

Chapter 6

Novel View Synthesis

The goal of view synthesis is to create intermediate views between the left and right images in a stereo pair. Given a disparity map, it is straightforward to relocate most of the pixels to the correct positions. However, there are some pixels that cannot be relocated, especially if they are being blocked in both the left and right views and are supposed to appear in the intermediate view. These occluded pixels are the most challenging problems in view synthesis. When they become disoccluded in one of the views, we have to artificially fill them. Fig. 6.1 illustrates an example of a double occlusion. The middle view is synthesized from the two original views and occlusion regions are highlighted in green. The blue rectangular areas indicate where the occlusion region exists in each view. Note that in the left view the bust covers the background region and in the right one the ring is occluding. We have no information as to what the contents of the disoccluded region should be, and so we must infer the pixel values based on neighboring regions.

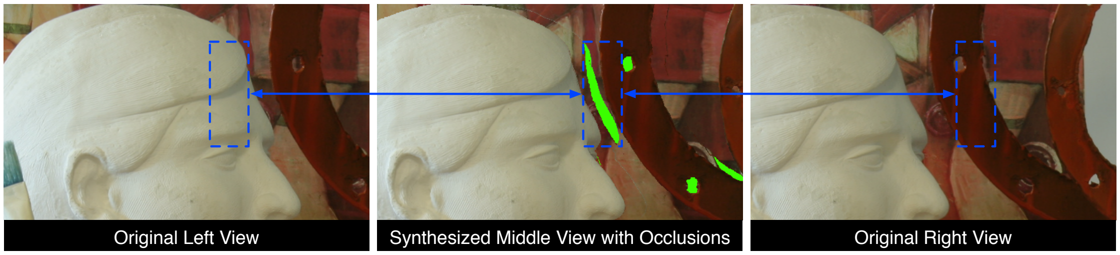


Figure 6.1: Illustrating occlusion in a synthesized view.

Numerous approaches have been undertaken to solve the view synthesis problem. An early approach to filling missing information (called occlusions) was view morphing, or the interpolation of pre-selected correspondence pixels from one image into the next. The work in [69] used a stereo geometry based approach to morph an arbitrary image into another. However, this approach required user-selected correspondences. Furthermore, morphing makes no assumptions about the 3D scene and therefore does not necessarily preserve the perspective transformation between the images. To make sure this is upheld (a condition necessary for multiview displays), others employed depth image based approaches. This preserves structure but reveals hidden regions in the background where foreground objects occlude them (the occlusion problem). Sparse reconstruction approaches, such as [70], can accurately reconstruct images with up to 50% of the pixels randomly removed because enough structure is preserved. Yet for images with a connected region of missing pixels, as is the case in a synthesized view, these methods fail. Not enough structural information exists in the region to fill it well. Several other approaches have been proposed, including methods based on inpainting [71] [72] or warping [73]. In our recent works [4, 5], we explored two methods for synthesizing views. The former places an emphasis on accuracy and a preservation of complex structure and texture, while the latter focuses on producing fast, visually-pleasing novel views. In the following sections, we summarize the two proposed approaches for view synthesis.

6.1 Initial Novel View Generation

To generate a new view from the existing stereo views, we may assign each pixel in the original view to a new location in the synthesized view. Under the rectified image assumption, point correspondences occur along horizontal scan lines. Consider the left and right cameras to be along a normalized baseline at positions 0 and 1, respectively, and let the virtual camera be at location α, where $0 < \alpha < 1$. We can map the pixels from the two views into the virtual view by horizontally shifting the pixel locations by the disparity of the pixel scaled by α

or $(1-\alpha)$ for the left and right views, respectively. More specifically, Let $f(x, y)$ denote a specific pixel in the original left view and $\delta(x, y)$ denote the corresponding disparity, then the pixel in the synthesized image $g(x, y)$ at position α is given by:

$$g(x - \alpha\delta(x, y), y) = f(x, y). \tag{6.1}$$

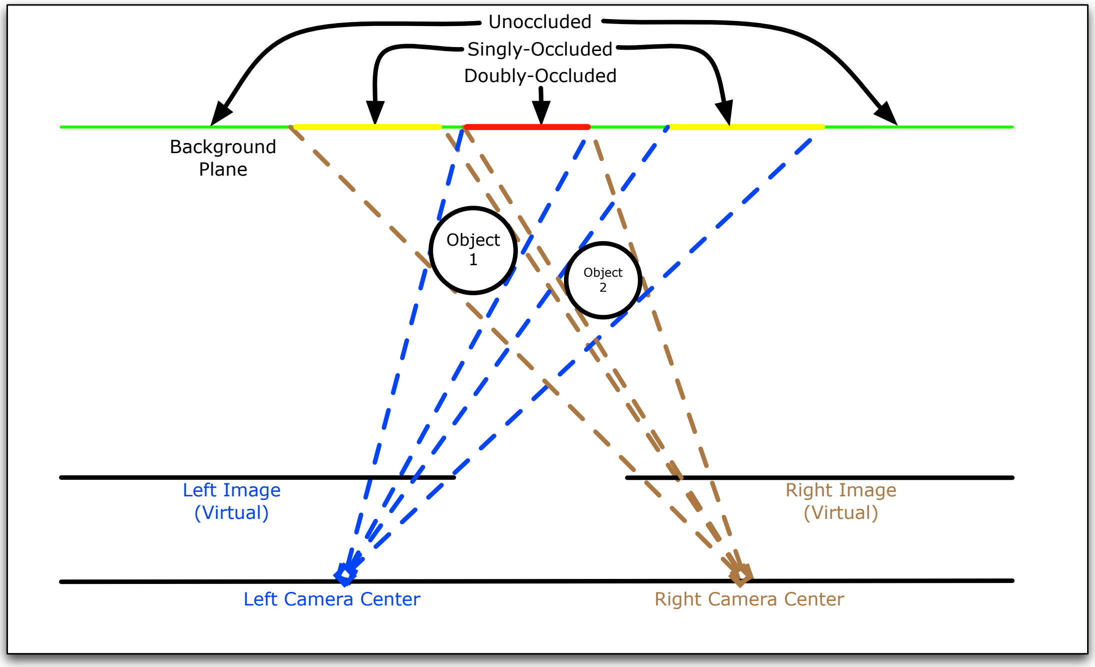


Figure 6.2: A top-down view of a 3D scene illustrating occlusions.

However, this mapping does not guarantee that every pixel of $g(x, y)$ gets a mapping. In other words, the range space of the mapping is smaller than the size of the image. Intuitively, the unmapped pixels are those occluded pixels. Since there is no information in these pixels, we need to fill them with information from their neighborhood and the layer to which they belong.

Pixels in the synthesized image can be one of three classes: *unoccluded*, *singly-occluded*, or *doubly-occluded* as illustrated in Fig. 6.2. The singly-occluded pixels are the simplest to handle. They are the ones which are hidden in one image but exist in the other. Since there is a one-to-one mapping in this case, the pixel simply gets assigned the only available value. Unoccluded pixels are those which have a mapping to a pixel in both original images (the many-to-one scenario). In these cases, we can either assign the pixel a value from one of the images (if camera properties are similar) or a weighted sum of the two images (if camera responses significantly differ). Lastly, we have the doubly-occluded pixels. These are the pixels which do not have a mapping from either image. This can happen when an objects in the foreground block an object in the background in both images. For this situation, we have no information and need to fabricate plausible pixel values that would appear visually pleasing to viewers and suitably maintain structure in the scene.

6.2 Background

Depth image-based rendering (DIBR) is commonly used to generate new virtual viewpoints for autostereoscopic displays. The three main steps of the DIBR framework are: preprocessing of the disparity maps, image warping, and hole filling. Assuming there exists a robust way to estimate the disparity maps, then the challenge of this method becomes the hole filling step, in which one must restore the occluded pixels in the new virtual view. Disocclusion refers to the process of recovering scene information obstructed by visible points and we refer to any occluded pixels that have been restored as disoccluded pixels.

In [74], a Gaussian filter is used to smooth the disparity map in the preprocessing step to eliminate disoccluded pixels. This method is easy to implement and computationally efficient. However, the synthesized images are unrealistic due to the visible geometric distortions, especially when the disoccluded region is large.

A better approach [71] combines depth-based-hole-filling and inpainting to restore the disoccluded pixels more accurately compared to prior inpainting methods without using depth information. While inpainting is a powerful tool to restore small disoccluded regions, it produces a notable amount of blur and can become computationally inefficient when the disoccluded region is large in the virtual view. Both of these methods described above produce visual artifacts as shown in [72] and degrade the 3D effect when the synthesized images are interlaced into a multiview image.

The work in [75] over segments the virtual view and pixels in segments connected to each disoccluded region are used with edge information to fill in the regions. The filling process is done by merging each disoccluded pixel to an attached segment and selecting a pixel in a neighboring segment to fill in that disoccluded pixel. This method fails when there is no strong edge or when complicated textures are present, making it difficult to merge disoccluded pixels to the correct segment. The method in [76] suggests generating two virtual views at the extreme left and right positions and uses view interpolation to generate the intermediate views. To enhance spatio-temporal consistency, the authors add disparity information to calculate priority that determines the filling order of occluded pixels. However, this method requires additional hole filling within each intermediate view.

6.3 Method 1: Spatially Consistent View Synthesis with Coordinate Alignment

Our first approach is a collaborative work that aims to explicitly maintain a relationship between the synthesized novel views. In this approach, we propose a novel method that uses coordinate alignment and background pixel extraction to synthesize highly accurate and spatially consistent intermediate views from a pair of stereo images and disparity maps. In contrast to the traditional depth image-based rendering (DIBR) methods, where useful background pixels are discarded in the warping process, the proposed method extracts these background pixels and uses them as candidates for an exemplar-based image inpainting technique (EBIIT) to synthesize realistic content in disocclusion regions. Our second contribution in this work is a coordinate alignment algorithm that aligns disocclusion regions in each view together and simultaneously synthesizes disocclusion regions to enhance spatial consistency across all virtual views.

In this work, we extend the work of [75] and propose a new method to synthesize consistent intermediate stereo images from a pair of stereo images that achieves high accuracy in quantitative metrics. We propose to warp the pixels and their coordinate indices from the reference view to the virtual views. We then align the disoccluded regions with those of the other virtual views based on their coordinate indices. The background layer is then extracted and EBIIT [77] is used to fill in the disocluded pixels. After the holes are filled, the disoccluded region is aligned back to the virtual views.

6.3.1 Approach

Pixel Classification

Each pixel in the virtual view is first classified in one of three categories: stable, unstable, or disoccluded. Stable pixels have only one pixel candidate and remain constant throughout the inference process. Unstable pixels have multiple pixel candidates and the candidate is selected that matches best with its neighboring pixels. Finally, disocclusion pixels have no pixel candidate and are occluded in both reference images. The candidates are obtained with pixel extraction and the disoccluded pixels are aligned and filled in with EBIIT.

Initial Virtual View Synthesis

As mentioned in Section 6.1, we may map the source views to the virtual view using the disparity maps. However, in this method, we need to be slightly more careful as we also wish to preserve the mappings for the future coordinate alignment step. In the initial step, color segments in the left I_L and right I_R reference images are extracted by [78]. In each segment, pixels with a disparity value that exceeds ± 20 from the mode are labeled as occluded under the assumption that disparity values vary smoothly on the surface of an object. The occluded pixels are then filled with the disparity of the nearest non-occluded neighboring pixels in the segment to generate the left and right refined disparity maps, D_L and D_R. After the initial disparity refinement step, a set of n virtual camera positions θ are defined. In our case, $n = 8$ since our goal is to generate the views for the 8-view Alioscopy display. Each $\theta_i \in \{0, 1\}$ is used to compute two disparity maps for virtual view i as

$$\mathbf{D}_{L,i} = \theta_i \mathbf{D}_L \quad \text{and} \quad \mathbf{D}_{R,i} = (1 - \theta_i)\mathbf{D}_R. \tag{6.2}$$

These disparity maps are used to generate placement matrices to warp pixels in the reference images to the virtual view. To evaluate the proposed method with the Middlebury [25, 79] dataset (which only provides 5 views), the three virtual cameras must be positioned at $\theta = \{1/4, 1/2, 3/4\}$ and the reference left and right cameras are positioned at 0 and 1 respectively. If we were synthesizing the views for the multiview display, however, we would need 8 views and therefore position six virtual cameras at $\theta = \{1/7, 2/7, 3/7, 4/7, 5/7, 6/7\}$.

Next, the disparity maps for each virtual view are used to compute the placement matrices that warp pixels from the reference view to the virtual view for the stable pixels with $\mathbf{P}^1_L$ and $\mathbf{P}^1_R$ and the unstable pixels with matrices $\mathbf{P}^2_L$ and $\mathbf{P}^2_R$. The advantage of warping pixels using placement matrices is that the refinement of small cracks and round-off errors is performed on the coordinates of the image to preserve all the texture of the virtual view.

These placement matrices are then used to warp pixels in the reference views to the virtual view i as

$$\mathbf{I}_i = \mathbf{I}_L(\mathbf{P}^1_{L,i} + \mathbf{P}^2_{L,i}) + \mathbf{I}_R(\mathbf{P}^1_{R,i} + \mathbf{P}^2_{R,i}) \tag{6.3}$$

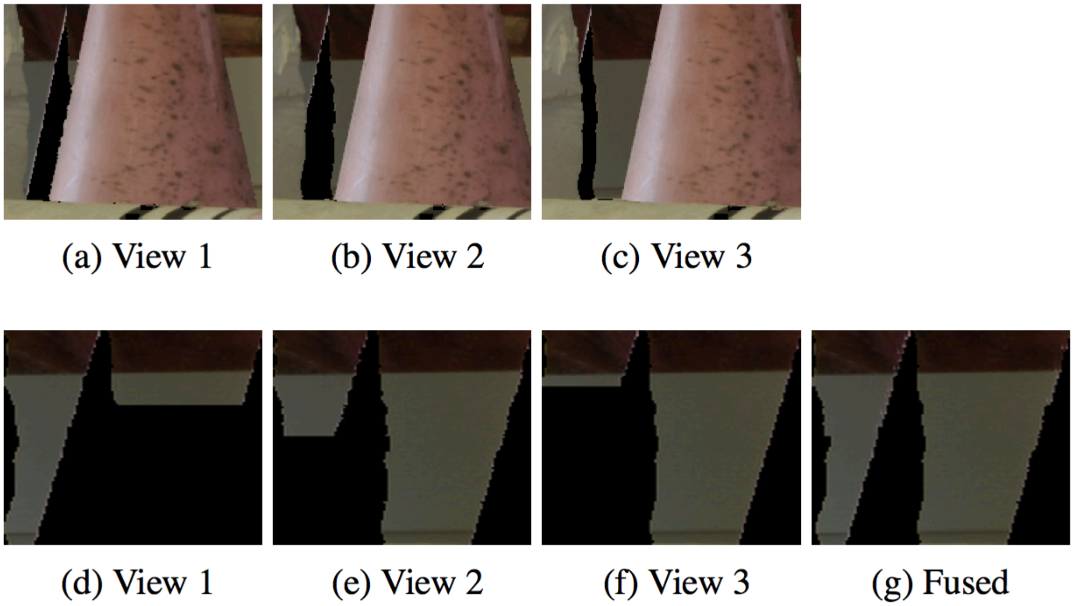
The pixel coordinate map is used to track the location of the disoccluded regions in the reference views and is computed as

$$\mathbf{C}_i = \mathbf{N}_L(\mathbf{P}^1_{L,i} + \mathbf{P}^2_{L,i}) + \mathbf{N}_R(\mathbf{P}^1_{R,i} + \mathbf{P}^2_{R,i}) \tag{6.4}$$

where each row of N is $[1, 2, \ldots, w]$, and w is the image width. Finally, the reference map is used to keep track of whether the pixel is warped from either the left or right view and is computed as

$$\mathbf{F}_i = 1(\mathbf{P}^1_{L,k} + \mathbf{P}^2_{L,k}) + 2(\mathbf{P}^1_{R,k} + \mathbf{P}^2_{R,k}). \tag{6.5}$$

As a result, the reference map $\mathbf{F}_i$ will contain a 1 indicating the left view was the source, 2 indicating the right view, or 0 if neither left nor right view map to that location (an occlusion).



(a) View 1 (b) View 2 (c) View 3

(d) View 1 (e) View 2 (f) View 3 (g) Fused

Figure 6.3: (a)-(c): Intermediate views after pixel warping. (d)-(f): Pixels extracted after alignment. (g) Fused image of (d), (e), and (f) used for hole filling.

Coordinate Alignment and Pixel Extraction for Spatial Consistency

Once all three initial virtual views, $\mathbf{I}_1$, $\mathbf{I}_2$, and $\mathbf{I}_3$, are synthesized using the placement matrices, each occluded region will appear in all three views but at different coordinate locations, as shown in the three images of the first row of Fig 6.3. To enforce consistency of the synthesized regions across all views, the proposed method aligns the coordinates of the disoccluded region onto the destination virtual view and backtracks these coordinates to the reference views to extract background neighboring pixels. These neighboring pixels will then be used to fill in the disoccluded pixels.

Aligning Coordinates to a Target View

We begin by partitioning each virtual view into n segments. When a disoccluded region appears in a segment, the corresponding virtual view, k, is selected as the destination view. The neighboring pixel coordinates are extracted for the selected disoccluded region to be synthesized. Then the coordinates from the other virtual views are aligned with the neighboring coordinates in the destination virtual view. We form the alignment matrix, which is an indicator matrix that determines whether pixel β in virtual view k is aligned to a pixel in virtual view l for an arbitrary row i, as follows

$$\mathbf{A}_{k,l,i}(\beta) = \begin{cases} 1 & \text{if} \quad \Phi(\beta) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

such that

$$\Phi(\beta) = (\mathbf{F}_k(\beta) - \mathbf{F}_l(\alpha^*)) + (\mathbf{C}_k(\beta) - \mathbf{C}_l(\alpha^*)), \tag{6.7}$$

where α^* is the pixel in view l that minimizes $\Phi(\beta)$. Intuitively, this means that if pixel α^* in view l has the appropriate coordinates that would map it to the same reference pixel as pixel β in view k, then these two pixels may be aligned for the hole filling step. This process is performed on all the boundary coordinates of the disoccluded region in view k. Once all the neighboring pixels are aligned, we may perform background pixel extraction to recover all of the background pixels needed for the hole filling step.

Pixel Extraction from Reference Images

The most common method for stereo-to-multiview conversion is to first warp the pixels to the virtual view and then to perform hole filling after all the pixels have been warped. This method removes useful information from the reference images to aid the hole filling process. To recover information from the reference images, the proposed method extracts pixels in the reference images that are neighboring pixels to the disoccluded pixels in the virtual view.

However the neighboring pixels to be extracted region might be obstructed in the virtual view. To extract the neighboring pixels, we use the reference indicator map to track which reference image was used to warp pixels to the virtual view. The background pixel extraction process for view k starts on the boundary of the disoccluded region and is extracted from the reference view as follows:

$$\mathbf{E}_k(x,y) = \begin{cases} 0 & \text{if} \quad \mathbf{F}_k(x,y) == 0 \\ \mathbf{I}_L(x, y - \mathbf{D}_{L,k}(x,y)) & \text{if} \quad \mathbf{F}_k(x,y) == 1 \\ \mathbf{I}_R(x, y + \mathbf{D}_{R,k}(x,y)) & \text{if} \quad \mathbf{F}_k(x,y) == 2 \end{cases} \tag{6.8}$$

This process continues until the disparity levels between neighboring pixels disagree. In Fig. 6.3, the first three images of the bottom row, (d), (e) and (f), show the background pixel extraction of a disoccluded region across 3 virtual views.

Fusion of Extracted Pixels

After the alignment and pixel extraction processes, the individual extracted regions are fused together to view k in this manner

$$\mathbf{I}_{fused} = \sum_{l \neq k}^{n} \mathbf{E}_l \mathbf{A}_{l,k} + \mathbf{E}_k \tag{6.9}$$

Fig. 6.3 (g) shows an example of a fused background pixel extraction. The proposed method compared to traditional DIBR not only provides more information but removes irrelevant objects to simplify the hole filling process of the disoccluded region. The alignment step only requires each occluded region to be synthesized once, independent of how many intermediate views one wishes to synthesize and also enhances spatial consistency across all intermediate views. In contrast to the conventional DIBR methods, where each view is synthesized independently, the proposed method only synthesizes one view.

Hole Filling

After the disoccluded region is processed by the proposed method, the image is partitioned into 5 × 5 patches and filled with the exemplar-based image inpainting technique [77] with a slight modification. We simplified the filling priority calculation of each patch center at x and y to

$$P(x,y) = \frac{|G_{=}(x,y)| + |G_{||}(x,y)|}{|G_{=}(x,y)G_{||}(x,y)| + \epsilon}, \tag{6.10}$$

where $G_{=}$ and $G_{||}$ are the horizontal and vertical gradients of the images and ϵ is a non-zero constant. After the patches are filled, the disoccluded region is shifted back to each of the virtual views as shown in Fig. 6.4.

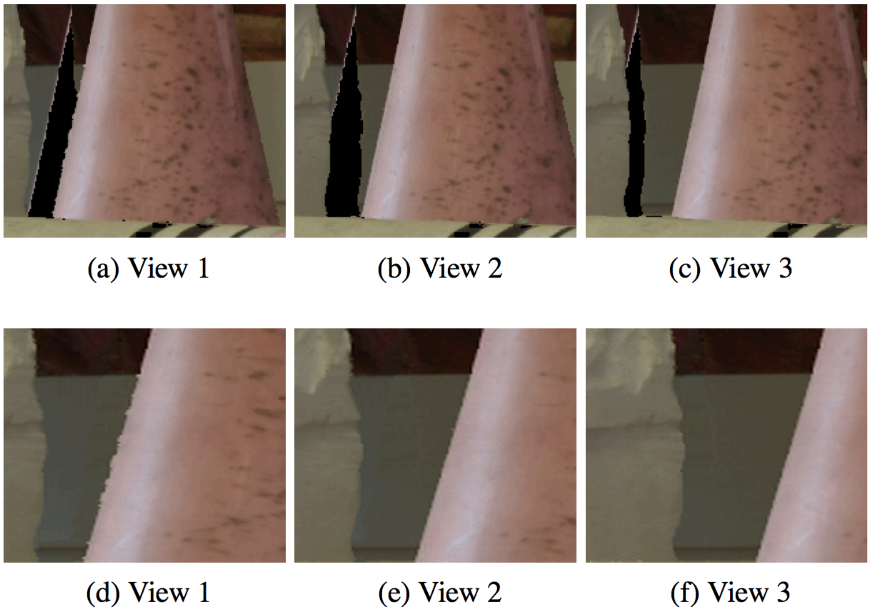

(a) View 1
(b) View 2
(c) View 3
(d) View 1
(e) View 2
(f) View 3

Figure 6.4: Row 1: Virtual views of camera positions 1, 2, and 3. A disoccluded region is labeled black and appears at three different coordinate locations. Row 2: Result of proposed method after hole filling.

6.4 Method 2: Efficient View Synthesis

Our second method is both fast and accurate. It consists of four main steps: generating an initial view, refining it, filling the holes in the disparity map, then filling holes in the color image.

6.4.1 Initial View Generation

The initial view generation is identical to that of the previous method, except that there is no need to retain the placement matrices in this case. We simply map all the existing pixels from the left and right views to the candidate virtual view position. If there is a case when a pixel is visible in both left and right images, we retain the pixel with the greater disparity value as measured from the candidate disparity maps because it is closer to the camera, and will occlude the pixel behind it.

6.4.2 Refinement

Disparity estimates along depth discontinuities are either missing from the original disparity maps or are unreliable [80]. This fact is manifested as ghost images and holes in the synthesized view. We tag these unreliable pixels by performing edge detection on a binary image showing the mapping of the original left and right views into the synthesized viewpoint. We implement edge detection using morphological operations for speed considerations (about 100 times faster than Canny edge detection). Therefore, a refinement map M is generated by

$$\mathbf{M} = ([\mathbf{I}_1 \oplus s)\ \mathbf{I}_1] \cup [(\mathbf{I}_2 \oplus s)\ \mathbf{I}_2]) \oplus s \tag{6.11}$$

where $\oplus$ denotes image dilation, $\mathbf{I}_1$ and $\mathbf{I}_2$ represent the binary map containing a 1 if the corresponding pixel was copied from the left or right image, respectively, and a 0 elsewhere, and the structuring element s is given by

$$s = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{6.12}$$

The two operands of the union in Equation 6.11 are the edge maps of the left and right mappings, and the result is dilated to increase the connectivity of edges and ensure that unreliable estimates are not left unfiltered. We filter the initial image and disparity map outputs with a 5×5 median filter at every location where $M = 1$. Not only does this refinement fix unreliable estimates while preserving reliable ones, it also dramatically decreases the amount of computation in future steps of the algorithm.

6.4.3 Filling the Disparity Map

Centered around each hole in the refined disparity map, we select an $N \times N$ neighborhood $\mathcal{N}$. Excluding the other hole pixels in the neighborhood, we form a histogram of B bins and also compute the variance $\sigma^2_{\mathcal{N}}$ of the disparities in $\mathcal{N}$. In the case when $\mathcal{N}$ contains no valid data (only holes), we increase N by $\triangle N$ and repeat. When a contiguous missing region belongs to a single disparity level, it makes sense to populate the holes with the most common disparity value. When multiple disparity levels cross a missing region, a lower disparity value (further from the camera) is preferable since it is more likely that a foreground object was occluding the background. We use the variance to discriminate between single versus multiple disparity levels. Thus, we seek to minimize the cost function

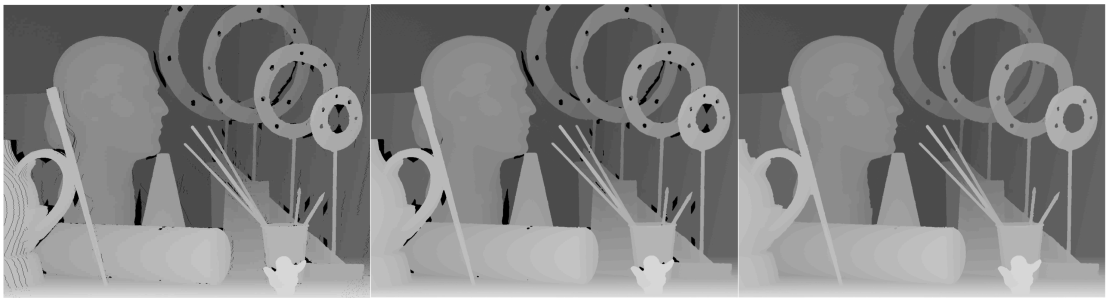
$$l(b_i) = \beta \sigma_{\mathcal{N}}^2 b_i + \frac{1}{c(b_i)}, \quad 1 \leq i \leq B \tag{6.13}$$

where b_i is the i-th bin of the histogram, $b_1 < \ldots < b_B$, $c(b_i)$ is the number of elements in b_i, and β is a tuning parameter. The β factor adjusts the balance between the two additive terms in 6.13. When the variance is high, the first term in 6.13 is emphasized since multiple disparity levels are likely present, and thus selecting the background level is preferable. When the variance is low, the first term in 6.13 is diminished since a single depth layer most likely surrounds the hole, making the maximum likelihood estimate of the (non-hole) data a better choice. Balancing the two terms by using the variance also provides robustness against noise. For instance, if the background disparity level were always selected, then even a few small values in the neighborhood could corrupt the estimate.

In our experiments, we used $N = 31$, $\triangle N = 12$, $B = 10$, and $\beta = 1000$. We selected b_i to be linearly spaced across the dynamic range of disparity values in each neighborhood. Fig. 6.5 shows the process of going from the initial synthesized disparity map to the final filled disparity map, which will be used in the hole filling process.

6.4.4 Filling the Synthesized View

Let $\mathcal{R}_i$ be the i-th contiguous region of missing pixels, and let the hole under consideration be x_{ij}, the j-th missing pixel in $\mathcal{R}_i$. Centered around x_{ij}, we select an $N \times N$ neighborhood $\mathcal{N}_{ij}$. We form a histogram of B bins and find the set of pixels that are in the same bin as the estimated disparity value for x_{ij} . The set of valid pixels $\mathcal{V}_{ij}$ in $\mathcal{N}_{ij}$ contains those pixels that are in the same disparity level as x_{ij} and are not holes. In the case when $\mathcal{V}_{ij}$ is empty, we increase the neighborhood size and repeat.



(a) Initial Disparity (b) Refined Disparity (c) Filled Disparity

Figure 6.5: Refinement and filling of disparity prior to filling the synthesized image.

We next form a super neighborhood $\mathcal{S}_i$, which is the smallest rectangle that contains all $\mathcal{N}_{ij}$. For this neighborhood, we perform k-means segmentation with C clusters on the grayscale neighborhood [81]. We denote the set of pixels in S_i assigned to the color class r as $\mathcal{K}_{ir}$ for $1 \leq r \leq C$. Note that this segmentation only needs to be performed once for each contiguous hole region.

Inaccuracies in the disparity map along borders of objects create spurious k-means classifications. Thus, we apply an 11×11 mode filter in the k-means domain around the pixels in $\mathcal{B}_i$, the border of $\mathcal{R}_i$. $\mathcal{B}_i$ is determined morphologically, similar to that for disparity estimation.

To determine the color class to which the hole x_{ij} belongs, we consider the valid border pixels in each color class:

$$\mathcal{C}_{ij;r} = \mathcal{V}_{ij} \cup \mathcal{B}_i \cup \mathcal{K}_{ir}, \quad 1 \leq r \leq C \tag{6.14}$$

We will assign a color class to x_ij based on the smallest median distance of $\mathcal{K}_{ir}$ to x_{ij} . In order to make a fair comparison of distance to each valid color border pixel, we choose the median over the same number of distances n:

$$n = min\{|\mathcal{C}_{ij;r}| : \mathcal{C}_{ij;r} \neq 0\} \tag{6.15}$$

Let $d_n(\mathcal{C}_{ij;r}, x_{ij})$ be the set of n smallest distances from each element in the valid border pixels to our current pixel. The color class assignment c is given by

$$c = \min_r \text{ median } d_n(\mathcal{C}_{ij;r}, x_{ij}) \tag{6.16}$$

In the case when no boundary pixels are valid, we let c be the most common color class among valid neighborhood pixels.

After determining the color class of x_ij , we estimate its value as the mean of the valid pixels belonging to color class c in the neighborhood. After filling all of the holes in this manner, we filter the edges of each contiguous missing region with a 5×5 median filter to denoise the border and produce the final synthesized view.

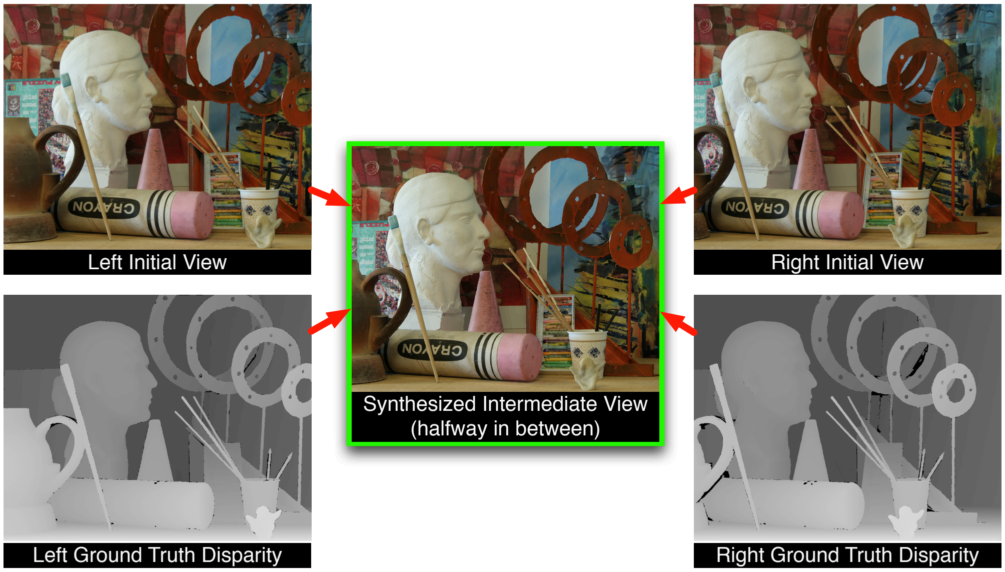

Left Initial View
Right Initial View
Synthesized Intermediate View
(halfway in between)
Left Ground Truth Disparity
Right Ground Truth Disparity

Figure 6.6: A good synthesis using ground truth disparity maps.

6.4.5 View Synthesis Dependence on Depth Estimates

A key component of view synthesis is accurate disparity estimation. Disparity informs us how pixels in one view map to the other view. Without accurate measurement of disparity, the task of synthesizing an artificial view becomes much more difficult. Fig. 6.6 illustrates a typical synthesis result of the intermediate view using ground truth disparity maps. In contrast, if poor disparity estimates are used as in Fig. 6.7, the final synthesized view will have many erroneous regions. When generating the multiview output, the errors are highly visible and in some cases overpower the sensation of the 3D effect altogether. Fig. 6.8 shows a close-up of the errors generated from the poor disparity estimates. In light of this strong dependence, it is of utmost importance that the disparity estimation algorithm be robust, especially when estimating the depth for a surgical setting.

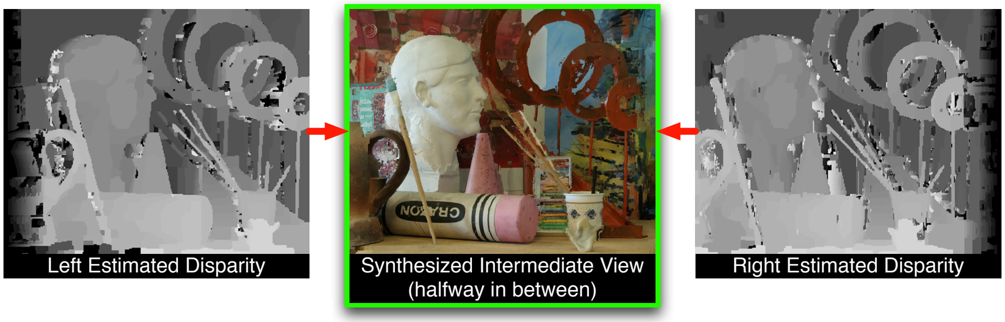


Figure 6.7: A bad synthesis using poorly estimated disparity maps.

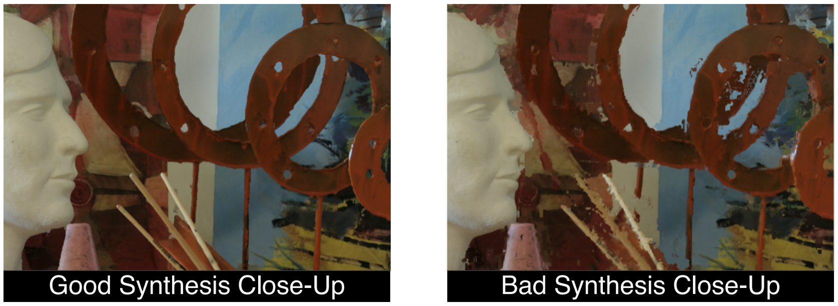


Figure 6.8: Highlighting the errors caused by bad disparity estimation.

Chapters 6 and 8, in part, are reprints of the material as it appears in [4, 5]. The dissertation author was a secondary investigator and author in these works.

Chapter 7

Real-time Computation

Let us review the main task of this dissertation. Our purpose is to build a complete system that will take live stereoscopic camera feeds and convert them to multiview with minimal latency and high accuracy. Fig. 7.1 depicts schematically what we are proposing. Given the stereoscopic sequence as input, we desire to map that input directly to a multiview sequence as output, which can then be multiplexed on an autostereoscopic display.

The process by which we do this is to first estimate initial frame-by-frame disparity maps, refine them using space-time minimization, and then synthesize the appropriate views. Yet, each of the steps in this approach has non-finite computational times. When they are all put together as a system, real-time computation is out of the question. Therefore, in this chapter, we discuss the modifications and innovations that we make in order to produce the real-time stereo-to-multiview system.

7.1 GPU Processing

With the realization that we need both high quality results and fast processing times, it became evident that using a graphics processing unit (GPU) for the entire stereo-to-multiview conversion process is crucial. Fortunately, the Compute Unified Device Architecture (CUDA) developed by NVIDIA [82] has vastly revolutionized the world of GPU programming in recent years, making parallel

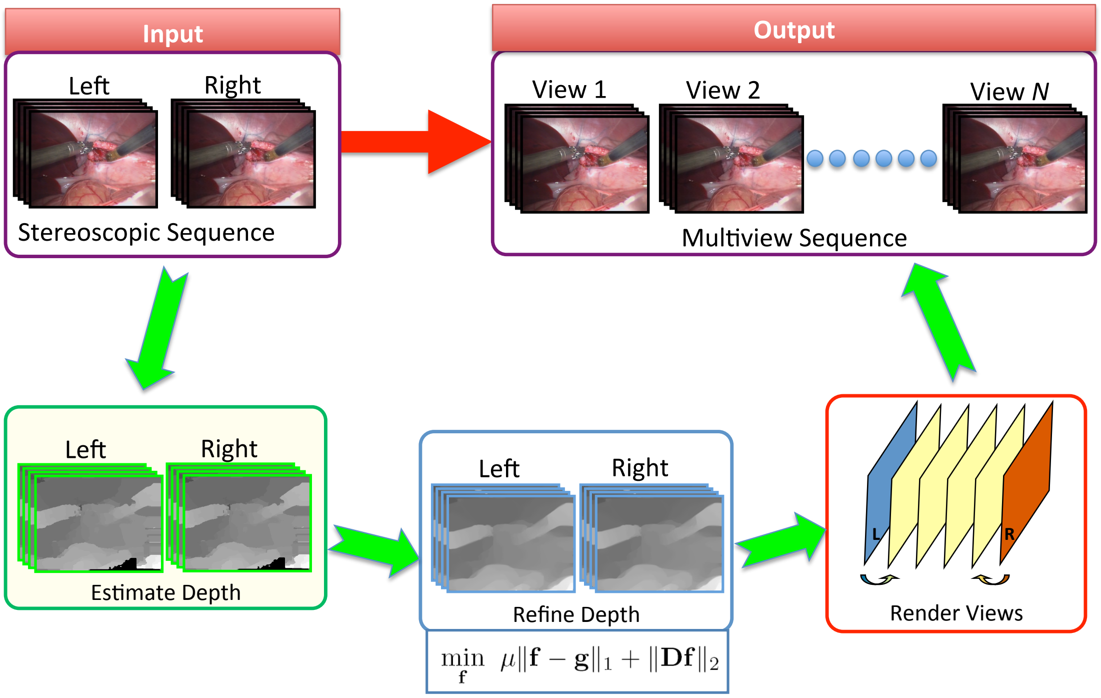


Figure 7.1: Illustrating the full stereo-to-multiview conversion pipeline.

programming easier than ever. The downside, however, is that parallel programming places tight constraints on programming style. CUDA distributes blocks randomly to idle streaming processor cores on the GPU to perform the computations. Every block has access to the global memory, but the read/write calls for the global memory are slow because it is uncached and, therefore, should be minimized. Each block also has its own shared memory and registers. Access times for shared memory and registers are very fast, but their size is limited. In addition to optimizing the algorithms for parallel processing, clever usage strategies for memory are crucial for maximizing speed. Most importantly, since all pixels are being processed in parallel, they must be treated independently of each other to maximize throughput. This is a huge challenge in stereo-to-multiview conversion where pixel interdependence is exploited to achieve high quality results in disparity estimation and the image warping process in view synthesis.

7.2 Rethinking Initial Disparity Estimation

In the previous chapters, we discussed various algorithms for the disparity estimation of images that can be broken down into two categories: local and global methods. We based our initial disparity estimation on HBP, which is a global method, using locally-adaptive weights. Programming a GPU with such methods is extremely difficult and sub-optimal. Additionally, our space-time minimization work is robust enough to improve arbitrary image-based disparity estimates. Therefore, we modify our initial disparity estimation step, replacing HBP with a local method and utilizing a robust matching cost.

7.2.1 Local Matching Cost Computation

The drawback with local methods is that disparity values are optimized over only a small neighborhood, rather than the entire image. For large, textureless regions, this can be problematic because the local method will equally weigh multiple correspondences as the right match. Therefore, we must modify the initial matching cost.

The Census Transform (CT) [83] is a non-parametric local transform. It relies on the relative ordering of local intensity values, instead of color intensity. Given an image, we first convert the color pixels to grayscale values, and construct a sliding window over each pixel. For each block, the census transform maps intensity values to a bit vector by performing a Boolean comparison between the center pixel intensity and its neighborhood pixels. If a neighboring pixel has a lower value than the center pixel, the bit is set to 0, otherwise the bit is set to 1. Taking $m = 3$ as an example, the following window is census transformed as

$$\begin{bmatrix} 149 & 160 & 230 \\ 153 & 154 & 156 \\ 156 & 157 & 152 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 & 1 \\ 0 & x & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{7.1}$$

Given two census transformed blocks (each of size $m \times m$), the Hamming distance, which is the logical exclusive-or (XOR) operation between the two blocks followed by the sum of non-zero entries, is used to determine the cost between them. For instance,

$$\text{hamming}\left\{\begin{bmatrix}0 & 1 & 1\\ 0 & x & 1\\ 1 & 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 0 & 0\\ 1 & x & 1\\ 1 & 1 & 0\end{bmatrix}\right\} = 3 \tag{7.2}$$

We can denote this matching cost as

$$\text{Cost}_{CT}(I_L(x), I_R(x)) = \text{hamming}(\text{census}(I_L(x)), \text{census}(I_R(x))). \tag{7.3}$$

The census transform is highly effective but will fail in low contrast regions. Therefore, we combine this cost with the sampling-insensitive absolute difference (BT) [84], which is used to compute the color cost between pixels by considering sub-pixels. For example, when computing the cost between two pixels $I_L(x,y)$ and $I_R(x,y)$, it calculates the following absolute color differences

$$\begin{aligned}\triangle I_{L,R} =& \quad |I_L(x,y) - I_R(x,y)|\\ \triangle I_{L^-,R} =& \quad |I_L(x-0.5,y) - I_R(x,y)|\\ \triangle I_{L^+,R} =& \quad |I_L(x+0.5,y) - I_R(x,y)|\\ \triangle I_{L,R^-} =& \quad |I_L(x,y) - I_R(x-0.5,y)|\\ \triangle I_{L,R^+} =& \quad |I_L(x,y) - I_R(x+0.5,y)|\end{aligned}$$

and selects the minimum. Birchfield and Tomasi [84] show that this can be done with little added computation to the standard absolute difference.

If we denote the BT cost as

$$\text{Cost}_{BT}(I_L(x), I_R(x)) = \text{BT}(I_L(x), I_R(x)), \tag{7.4}$$

then we can combine both CT and BT costs into a single distance function for matching pixel $p = I_L(x)$ with pixel $q = I_R(x)$ as

$$\text{Dist}(p,q) = \frac{1}{2}\left((1 - exp(-\tfrac{\text{Cost}_{CT}(p,q)}{\lambda_{CT}}) + (1 - exp(-\tfrac{\text{Cost}_{BT}(p,q)}{\lambda_{BT}})\right). \tag{7.5}$$

7.2.2 Cost Aggregation

While the combined CT-BT matching cost is very robust, it lacks the accuracy levels seen with a fully global method. Therefore, to further enhance quality, we perform an aggregation step based on the method of [85]. The result of the distance function is a three-dimensional map of errors where each error value is calculated independently. While it is possible to infer the disparity (at each pixel) by picking the disparity that return the smallest error, *i.e.*,

$$\mathbf{D}(x, y) = \underset{d}{\operatorname{argmin}} \ \epsilon(x, y, d), \tag{7.6}$$

where $\epsilon(x, y, d)$ is the error value for the pixel located at (x, y) for the chosen label d, the result can be noisy as each pixel is treated independently of its neighbors. Therefore, we apply the cross-based aggregation method to enhance spatial smoothness.

Without loss of generality, we focus on the left image and denote it as $I(\mathbf{x})$. At each color pixel $\mathbf{x}_p = (x_p, y_p)$, we want to define a neighborhood $U(\mathbf{x}_p)$ such that for all pixels in $U(\mathbf{x}_p)$, the colors are similar. To this end, we first define the top margin of $U(\mathbf{x}_p)$ to be the farthest vertical pixel $\mathbf{x}_q$ such that the color difference is less than a threshold τ:

$$\mathbf{v}_p^+ = \min_{\mathbf{x}_q}\{\mathbf{x}_q| \ ||I(\mathbf{x}_p) - I(\mathbf{x}_q)||_\infty < \tau, \mathbf{x}_q \in \text{positive vertical direction}\}$$

where $||\cdot||_\infty$ is the maximum of the three color components of $I(\mathbf{x})$. Similarly, we define the bottom margin as

$$\mathbf{v}_p^- = \min_{\mathbf{x}_q}\{\mathbf{x}_q| \ ||I(\mathbf{x}_p) - I(\mathbf{x}_q)||_\infty < \tau, \mathbf{x}_q \in \text{negative vertical direction}\}$$

The top and bottom margins define a vertical strip of pixels labeled as $\{\mathbf{v}_p^-, \ldots, \mathbf{v}_p^+\}$. We repeat the same process horizontally to get:

$$\begin{aligned}
\mathbf{h}_q^+ &= \min_{\mathbf{x}_r}\{\mathbf{x}_q| \ ||I(\mathbf{x}_q) - I(\mathbf{x}_r)||_\infty < \tau, \mathbf{x}_r \in \text{positive horizontal direction}\} \\
\mathbf{h}_q^- &= \min_{\mathbf{x}_r}\{\mathbf{x}_q| \ ||I(\mathbf{x}_q) - I(\mathbf{x}_r)||_\infty < \tau, \mathbf{x}_r \in \text{negative horizontal direction}\}
\end{aligned}$$

Fig. 7.2 illustrates graphically what the cross-based aggregation is doing. Note that the horizontal margins are defined for each $\mathbf{x}_q$ in the vertical strip. Thus, there is a set of horizontal strips for each vertical strip. The union of all these strips defines the neighborhood $U(\mathbf{x}_p)$ (denoted with the red boundary in the figure).

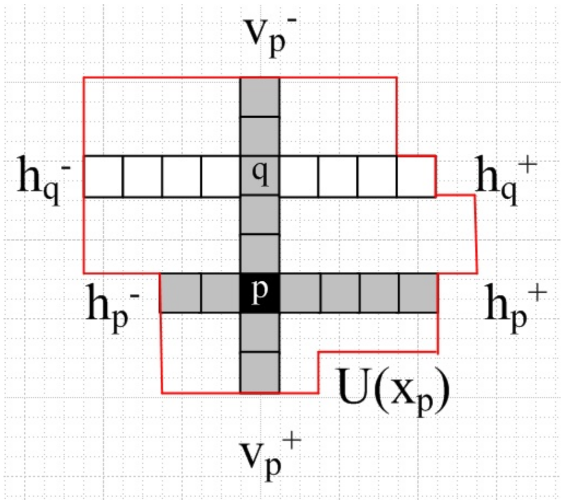


Figure 7.2: Illustration of Cross-based Aggregation.

With the aid of cross-based aggregation, we perform a non-uniform average of the three-dimensional error map. Specifically, for each pixel location, we take the average of error values within the neighborhood $U(\mathbf{x}_p)$ to form the cost volume:

$$C(\mathbf{x}_p, d) = \frac{1}{|U(\mathbf{x}_p)|} \sum_{\mathbf{x}_q \in U(\mathbf{x}_p)} \epsilon(\mathbf{x}_q, d) \tag{7.7}$$

where $|U(\mathbf{x}_p)|$ denotes the cardinality of the set $U(\mathbf{x}_p)$. The disparity can then be determined by picking the value that minimizes this cost at each pixel location $\mathbf{x}_p$:

$$\mathbf{D}(x, y) = \underset{d}{\operatorname{argmin}} \; C(\mathbf{x}_p, d). \tag{7.8}$$

7.3 Real-time Space-time Minimization

Once we have our frame-by-frame disparity estimates, we still need to make sure that they are spatio-temporally consistent. The method we have proposed thus far, however, although very fast, is a batch process. As a reminder, we must form a three-dimensional space-time volume by stacking N separate frames of the video on top of each other. This means that we will at best have a latency equal to the number of batch frames, N. This is unacceptable for a surgical setting, as the delay will be immediately noticeable for anything more than just a few frames.

We modify the original minimization of

$$\underset{\mathbf{f}}{\operatorname{argmin}}\ \mu||\mathbf{f}-\mathbf{g}||_1 + ||\mathbf{D}\mathbf{f}||_2, \tag{7.9}$$

to the real-time TV/L1 problem as an optimization problem in the form of

$$\underset{\mathbf{f}}{\operatorname{argmin}}\ \mu||\mathbf{f}-\mathbf{g}||_1 + ||\mathbf{D}\mathbf{f}||_2 + \mu\kappa||\mathbf{B}\mathbf{f}-\hat{\mathbf{f}}||_1 \tag{7.10}$$

which can also be written as

$$\underset{\mathbf{f}}{\operatorname{argmin}}\left\|\begin{pmatrix} \mu\mathbf{I} \\ \mu\kappa\mathbf{B} \end{pmatrix}\mathbf{f} - \begin{pmatrix} \mu\mathbf{g} \\ \mu\kappa\hat{\mathbf{f}} \end{pmatrix}\right\|_1 + ||\mathbf{D}\mathbf{f}||_2. \tag{7.11}$$

Here, $\hat{\mathbf{f}}$ is the result from the previous frame and $\mathbf{B}$ is a diagonal matrix whose (i,i)-th element is 1 if there is no motion at pixel i, and 0 otherwise. All other parameters are the same as before.

The problem can be rewritten as:

$$\underset{\mathbf{f}}{\operatorname{argmin}}\ \mu||\mathbf{A}\mathbf{f}-\mathbf{b}||_1 + ||\mathbf{D}\mathbf{f}||_2 \tag{7.12}$$

where $\mathbf{A}$ is $\begin{pmatrix} \mathbf{I} \\ \kappa\mathbf{B} \end{pmatrix}$ and $\mathbf{b}$ is $\begin{pmatrix} \mathbf{g} \\ \kappa\hat{\mathbf{f}} \end{pmatrix}$.

We can then form the equivalent constrained problem as before with

$$\underset{\mathbf{f}}{\operatorname{argmin}}\ \mu||\mathbf{r}||_1 + ||\mathbf{u}||_2 \tag{7.13}$$

subject to the constraints that $\mathbf{r} = \mathbf{A}\mathbf{f} - \mathbf{b}$ and $\mathbf{u} = \mathbf{D}\mathbf{f}$ and the augmented Lagrangian function is then given by

$$\begin{aligned} L(\mathbf{f},\mathbf{r},\mathbf{u},\mathbf{y},\mathbf{z}) = \mu||\mathbf{r}||_1 + ||\mathbf{u}||_2 - \mathbf{z}^T(\mathbf{r}-\mathbf{A}\mathbf{f}+\mathbf{b}) + \frac{\rho_o}{2}||\mathbf{r}-\mathbf{A}\mathbf{f}+\mathbf{b}||_2^2 \\ - \mathbf{y}^T(\mathbf{u}-\mathbf{D}\mathbf{f}) + \frac{\rho_r}{2}||\mathbf{u}-\mathbf{D}\mathbf{f}||_2^2 \end{aligned} \tag{7.14}$$

We use the same alternating direction method to find a saddle point of the augmented Lagrangian as we did previously. At the k-th iteration, we solve the following sub-problems sequentially.

7.3.1 The ‘f’ sub-problem

The *‘f’ sub-problem* can be written as

$$\underset{\mathbf{f}}{\operatorname{argmin}}\ \mathbf{z}^T\mathbf{A}\mathbf{f} + \frac{\rho_o}{2}||\mathbf{r} - \mathbf{A}\mathbf{f} + \mathbf{b}||_2^2 + \mathbf{y}^T\mathbf{D}\mathbf{f} + \frac{\rho_r}{2}||\mathbf{u} - \mathbf{D}\mathbf{f}||_2^2 \tag{7.15}$$

This function is fully differentiable and so we can formulate the normal equations as

$$(\rho_o\mathbf{A}^T\mathbf{A} + \rho_r\mathbf{D}^T\mathbf{D})\mathbf{f} = \rho_o\mathbf{A}^T(\mathbf{b} + \mathbf{r}) - \mathbf{A}^T\mathbf{z} + \mathbf{D}^T(\rho_r\mathbf{u} - \mathbf{y}) \tag{7.16}$$

Expressing this explicitly using:
$\mathbf{D} = [\mathbf{D}_x^T\mathbf{D}_y^T]^T$, $\mathbf{r} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{pmatrix}$, $\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix}$, $\mathbf{A} = \begin{pmatrix} \mathbf{I} \\ \kappa\mathbf{B} \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} \mathbf{g} \\ \kappa\hat{\mathbf{f}} \end{pmatrix}$,
we arrive at the following normal equations

$$\begin{aligned}(\rho_o(1 + \kappa^2\mathbf{B}^T\mathbf{B}) + \rho_r\mathbf{D}^T\mathbf{D})\mathbf{f} = \rho_o(\mathbf{g} + \kappa^2\mathbf{B}^T\hat{\mathbf{f}} + \mathbf{r}_1 + \kappa\mathbf{B}^T\mathbf{r}_2) - (\mathbf{z}_1 + \kappa\mathbf{B}^T\mathbf{z}_2)\\ + \mathbf{D}_x^T(\rho_r\mathbf{u}_x - \mathbf{y}_x) + \mathbf{D}_y^T(\rho_r\mathbf{u}_y - \mathbf{y}_y)\end{aligned} \tag{7.17}$$

This can be solved using the method we previously described in three steps: 1) computing the Fourier transform, 2) performing element-wise division, and 3) computing the inverse Fourier transform.

7.3.2 The ‘u’ sub-problem

The *‘u’ sub-problem* is identified as

$$\underset{u}{\operatorname{argmin}}\ ||\mathbf{u}||_2 - \mathbf{y}^T\mathbf{u} + \frac{\rho_r}{2}||\mathbf{u} - \mathbf{D}\mathbf{f}||_2^2 \tag{7.18}$$

and the solution is readily given by the shrinkage formula as

$$\mathbf{u}_x = \max\left\{\sqrt{|\mathbf{v}_x^2| + |\mathbf{v}_y^2|} - \tfrac{1}{\rho}, 0\right\}\frac{\mathbf{v}_x}{\sqrt{|\mathbf{v}_x^2| + |\mathbf{v}_y^2|}} \tag{7.19}$$

$$\mathbf{u}_y = \max\left\{\sqrt{|\mathbf{v}_x^2| + |\mathbf{v}_y^2|} - \tfrac{1}{\rho}, 0\right\}\frac{\mathbf{v}_y}{\sqrt{|\mathbf{v}_x^2| + |\mathbf{v}_y^2|}} \tag{7.20}$$

where $\mathbf{v}_x$ and $\mathbf{v}_y$ are given by

$$\mathbf{v}_x = \mathbf{D}_x\mathbf{f} + \frac{1}{\rho}\mathbf{y}_x \tag{7.21}$$

$$\mathbf{v}_y = \mathbf{D}_y\mathbf{f} + \frac{1}{\rho}\mathbf{y}_y \tag{7.22}$$

7.3.3 The 'r' sub-problem

The *'r' sub-problem* is given by

$$\underset{r}{\text{argmin}} = \mu||\mathbf{r}||_1 - \mathbf{z}^T\mathbf{r} + \frac{\rho_o}{2}||\mathbf{r} - \mathbf{Af} + \mathbf{b}||_2^2 \tag{7.23}$$

and it can also be solved in the same manner as the *'u' sub-problem*, yielding the solution

$$\mathbf{r} = \max\left\{|\mathbf{Af} - \mathbf{b} + \tfrac{1}{\rho_o}\mathbf{z}| - \tfrac{\mu}{\rho_o}, 0\right\}\text{sign}(\mathbf{Af} - \mathbf{b} + \frac{1}{\rho_o}\mathbf{z}) \tag{7.24}$$

7.3.4 Lagrange multiplier update parameters

Finally, the Lagrange multipliers are updated as

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \rho_r(\mathbf{u}^{k+1} - \mathbf{Df}^{k+1}) \tag{7.25}$$

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \rho_o(\mathbf{r}^{k+1} - \mathbf{Af}^{k+1} + \mathbf{b}) \tag{7.26}$$

The optimization of each of the sub-problems is iterated until convergence of the termination criteria or a certain fixed number of iterations have completed.

7.4 Extension to Multiple GPUs

The proposed system is implemented on dual NVIDIA GTX580 GPUs, and achieves frame-rates above $30fps$ for the surgical data. The entire system is implemented in CUDA using the GPU by exploiting parallelism. We split the workload between the GPUs vertically so that each GPU handles half of the image every frame. However, the aggregation and space-time minimization steps require apron pixels because neighboring pixels are considered when performing the processing. Therefore, each GPU also receives a small number of additional apron pixels from the other view which are only used in the processing to reduce error in the border regions. When the final synthesized view is generated, the apron pixels are discarded.

The method is fully scalable and can be extended to any number of GPUs or even multiple processing clusters if the stereo-to-multiview process is being run on extremely large datasets.

Chapter 8

Algorithmic Evaluation

We begin by evaluating our disparity estimation algorithm. First, we examine its performance on images, particularly how we may apply our spatio-temporal denoising to improve the results of image-based algorithms. We then examine the results on synthetic video sequences with ground truth disparity maps for a quantitative evaluation, and show some results with real video sequences as well as a surgical sequence as a qualitative validation of the results. Afterwards, we examine the efficacy of our view synthesis methods to validate that quantitatively going from stereo to multiview is a feasible task, particularly for surgery.

8.1 Disparity Estimation Evaluation

Since our method of disparity estimation is applicable to both images and videos, we tested it on both to evaluate its performance. For both types of data, we first performed our initial disparity estimation, followed by the total variation minimization to enforce consistency. As mentioned previously, for images we set the size of the space-time volume to a single frame, so that the minimization would only enforce spatial consistency, as no additional frames were available.

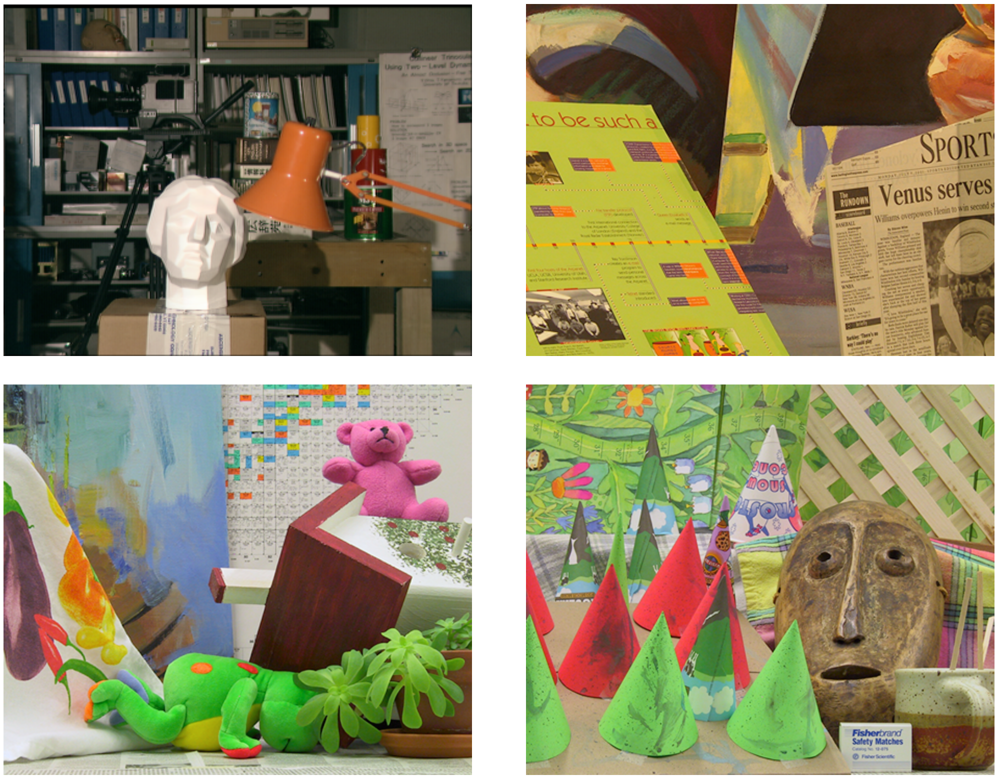
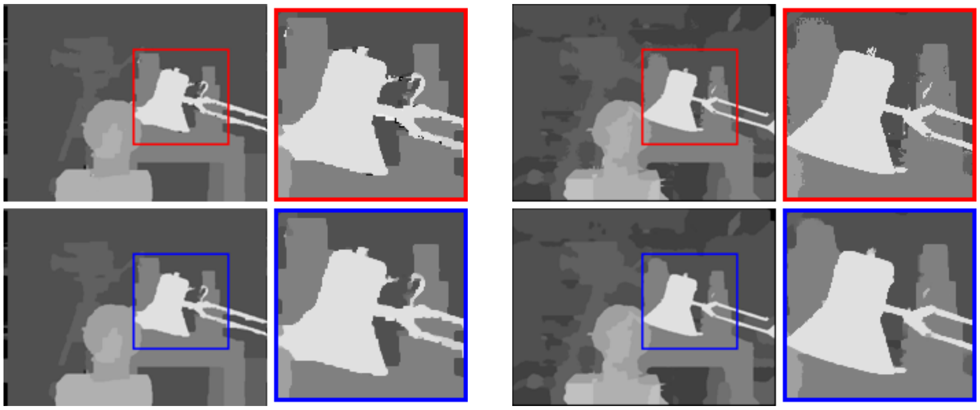


Figure 8.1: The four Middlebury evaluation stereo pairs (only left view shown). Clockwise from top left: Tsukuba, Venus, Teddy, Cones.

8.1.1 Image Results

The first experiment aims at evaluating the performance of the algorithm for static images. We set $\beta_x = \beta_y = 1$ and $\beta_t = 0$ in Eq. (5.7), indicating no contribution from a temporal component. Four evaluation datasets from Middlebury (see Fig. 8.1) are used to test our performance on images. At the time of our experiments, there were 99 methods on the Middlebury website, even though more algorithms have been submitted since then. For all of the 99 top-ranked methods submitted to the website, the space-time minimization algorithm is applied to their outputs in order to refine their results.

Our proposed algorithm consists of an initial disparity estimation step and a post-processing refinement step. It should be noted that the initial disparity estimation steps performed by the 99 methods are different due to the methodologies they use. As a result, the run time, memory requirement, amount of bad



(a) Algorithm No. 8

(b) Algorithm No. 78

Figure 8.2: Image disparity refinement on algorithms 8 and 78 (randomly chosen) from Middlebury for "Tsukuba". Red box: Before applying the proposed method. Blue box: After applying the proposed method.

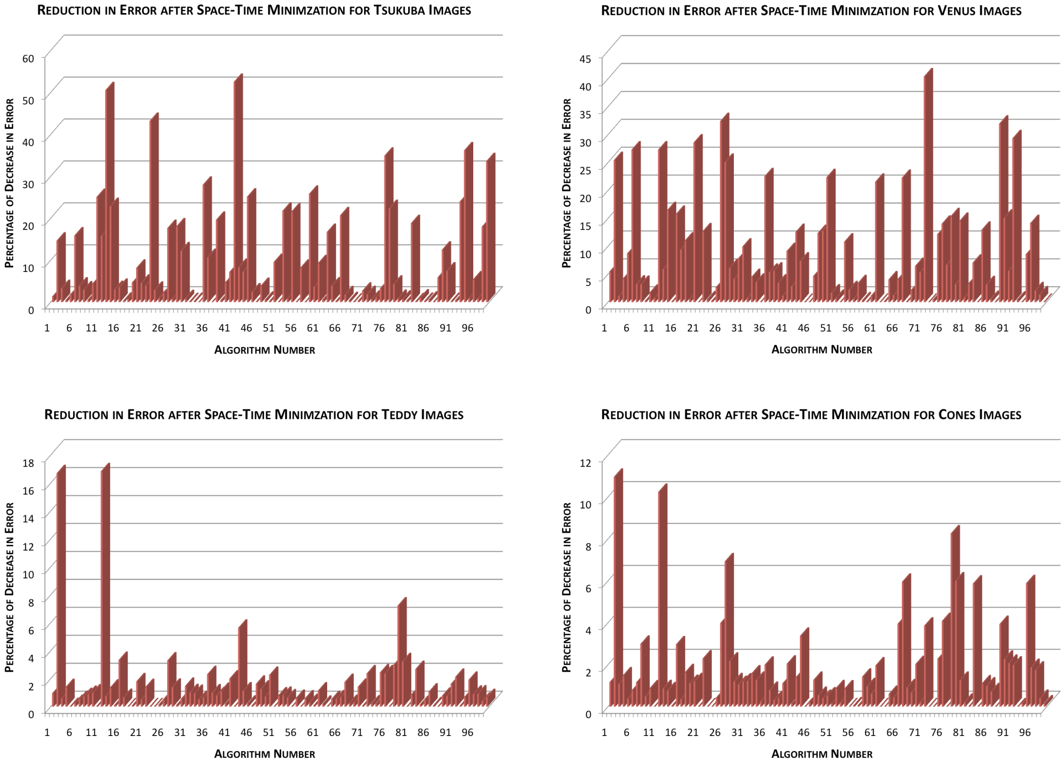


Figure 8.3: Percentage error reduction (in terms of number of bad pixels) by applying the proposed algorithm to all methods on the Middlebury database.

pixels and spatial smoothness for the initial estimations are all different. In the post-processing refinement step, however, the space-time minimization algorithm is *independent* of the initial estimation in the sense that it is applied to the *outputs* of the initial methods. The run times and memory requirements in this step are similar for all 99 methods. The goal of this experiment is to show the measurable improvement when our space-time minimization algorithm is applied to images.

Fig. 8.2 illustrates results before and after our method for two randomly chosen algorithms, and Fig. 8.3 shows the percentage of error reduction by applying the proposed algorithm to all methods on the Middlebury database. The higher bars in the plots indicate that the proposed algorithm reduces the error by a greater amount. The metric for error measurement is the percentage of bad pixels (or pixels having an incorrect disparity value by a difference greater than a threshold of 1).

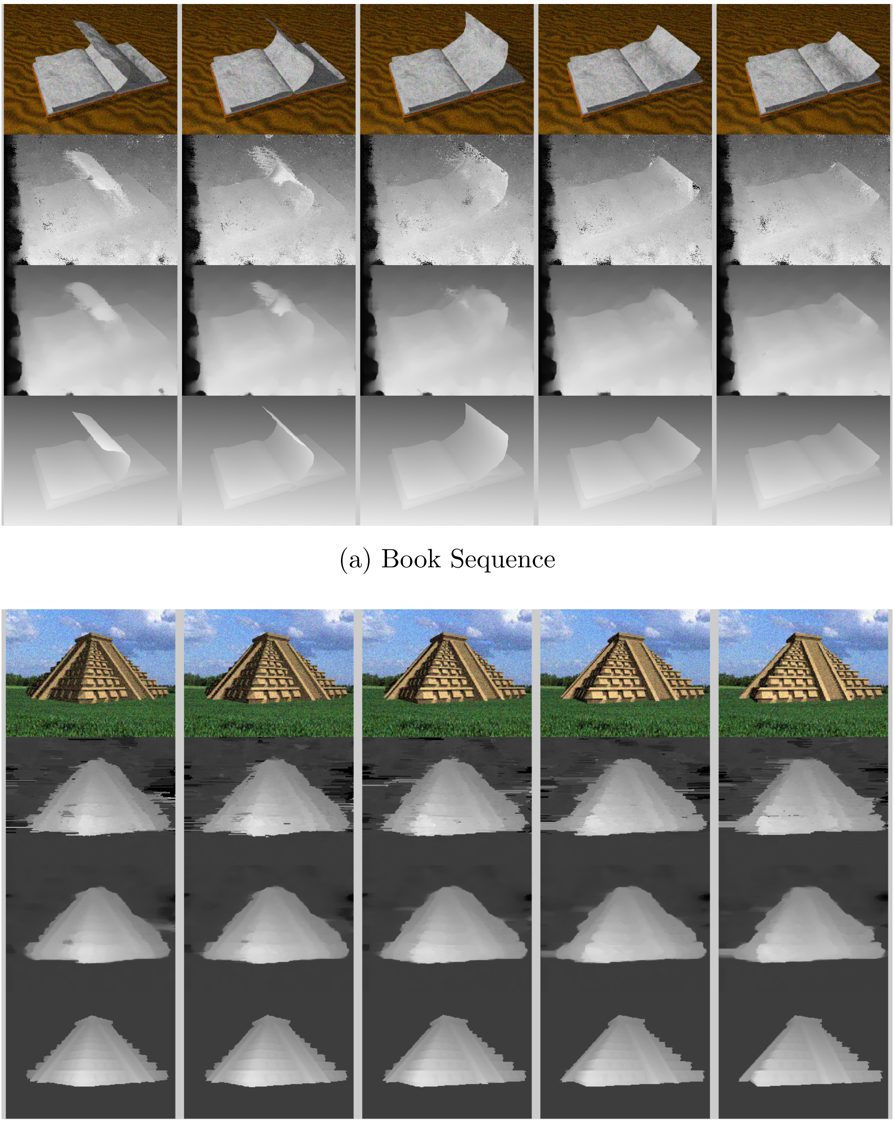
As shown in the plots, it can be observed that for the Tsukuba and Venus datasets, the errors are typically reduced by a large margin of over 10%. For the other two datasets, though the error reduction margin is smaller, we still show general improvement. While there is less error reduction for some datasets, it is important to note that error reduction is always non-negative. In other words, we never make the initial disparity estimates worse by applying our denoising method. Furthermore, for every single algorithm, we provide improvement in at least one of the image sets.

8.1.2 Video Results

Synthetic Data

Due to the lack of existing stereo sequences with ground truth disparity maps, effective evaluation of video disparity estimation techniques has been limited. In this subsection, we show a quantitative evaluation based on a synthetic dataset generated by Richardt *et al.* [35]. The dataset consists of five stereo sequences with associated ground truth disparity maps.

Fig. 8.4 shows five consecutive frames of the sequences "Book" and "Temple." For each video, we show the original left frames, initial disparities estimated using TDCB [35], refinements with our proposed algorithm, and ground truth. It



(a) Book Sequence

(b) Temple Sequence

Figure 8.4: Disparity refinement for synthetic sequences. For each video, we have: Top row: Original. Second row: Initial disparity from [35]. Third row: Proposed method with $\mu = 0.75$ and $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$. Bottom row: Ground truth.

can be seen that even by visual inspection the resultant disparity maps are more spatio-temporally consistent. In fact, the percentage of bad pixels per frame of each sequence is reduced significantly, as shown in Table 8.1.

To validate the robustness of our method, we adopt the same approach Richardt *et al.* used in [35] by incorporating additive Gaussian noise with a σ ranging from 0 to 100 (0 to 0.392 for intensities normalized to 1) to the stereo

Table 8.1: Versatility of TV on various methods. Refinements have "-TV" as a postscript. Values are the average percent of bad pixels (threshold of 1).

	Sequences				
	Book	Street	Tanks	Temple	Tunnel
TDCB-TV	**27.10**	**17.45**	**23.25**	**21.94**	**32.21**
TDCB [35]	38.95	24.17	29.34	29.89	33.01
DCB-TV	**35.31**	**22.45**	**23.00**	**27.38**	**22.41**
DCB [35]	47.24	30.91	33.56	37.59	24.04
DCB2-TV	**48.66**	**31.91**	**41.28**	**32.14**	**30.43**
DCB2 [35]	53.92	38.02	45.67	40.97	31.19

video. The results are shown in Fig. 8.5. Again, the proposed method nearly always provides a lower bound to the error of the initial disparity estimates, which verifies the robustness of the proposed method.

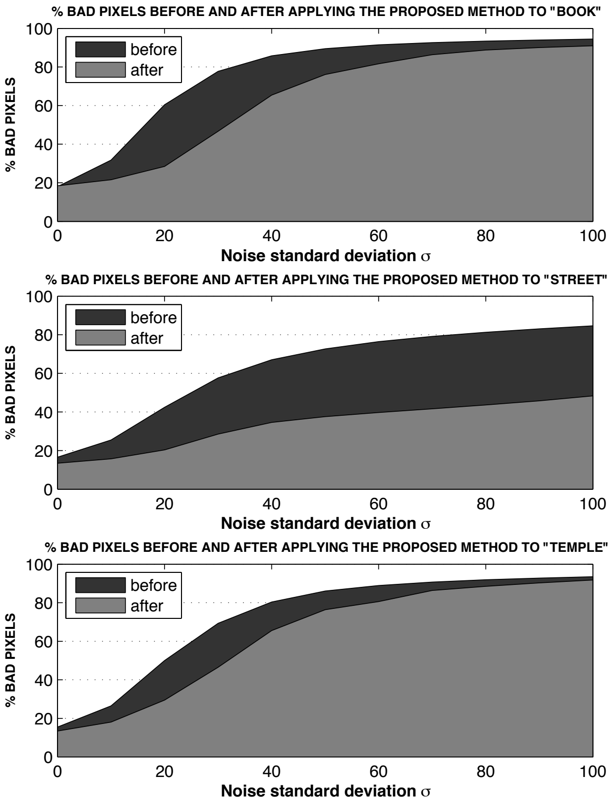
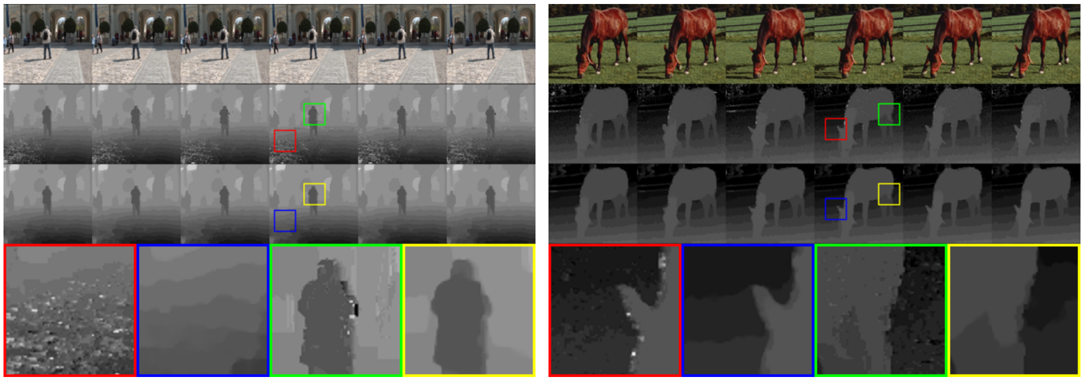


Figure 8.5: Percentage of bad pixels as a function of Gaussian noise before and after applying the proposed method to sequences "Book", "Street", and "Temple".

Real Data

The drawback of most algorithms is that they are limited to only specific datasets, often rendered with simplifying assumptions to aid development. We desired our method, however, to robustly work on any video sequence. Therefore, the proposed algorithm is tested for two real stereo video sequences: "Old Timers" and "Horse" [86]. The initial disparity is estimated using our image-based method, a variant of HBP [29] with locally-adaptive support weights [36] and refined using the space-time minimization method (with $\mu = 0.75$, $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$). Fig. 8.6a shows six frames of a zoomed-in region of the left view from the "Old Timers" sequence. Notice that the image-based results contain both spatial noise and temporal inconsistencies, particularly in the background area to the right of the male figure. After refinement, these errors are removed while object edges are still preserved. Another sequence "Horse" is shown in Fig. 8.6b, indicating similar improvements.



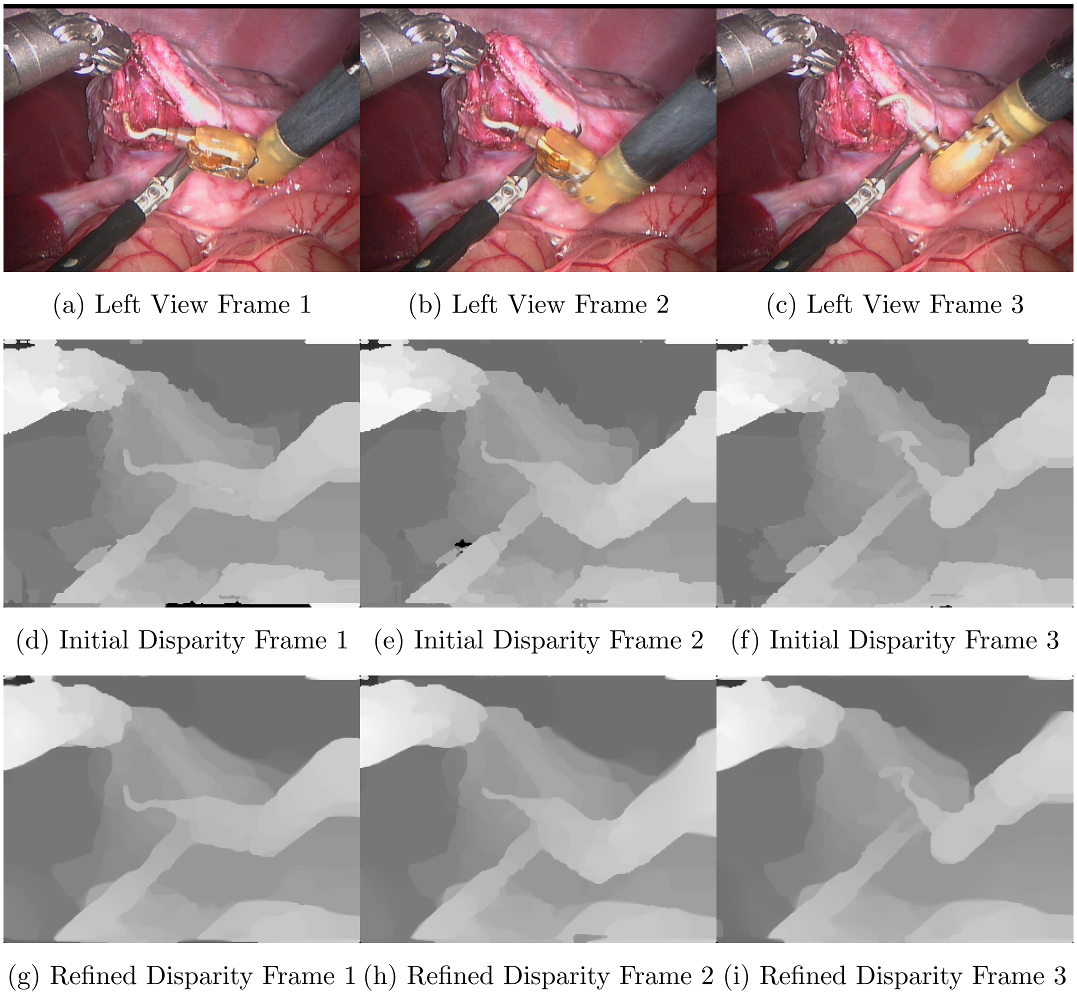
(a) Old Timers

(b) Horse

Figure 8.6: Results for real videos. First row: Left view of the stereo video. Second row: Initial disparity estimate. Third row: Refinement using the proposed method with parameters $\mu = 0.75$ and $(\beta_x, \beta_y, \beta_t) = (1, 1, 2.5)$. Fourth row: Zoomed-in results.

Surgical Data

Fig 8.7 shows results for disparity estimation on real surgical footage. As can be seen from the first row, it is much more difficult to discern depth in such a sequence than in natural scenes because many of the normal monocular cues do not exist here (texture gradients, shading, relative size, etc.). However, our algorithm performs well even in such difficult sequences. In the second row, we show the corresponding disparity maps computed with the image-based method (no spatio-temporal consistency). In the final row, our full spatio-temporally consistent disparity estimation is performed. Upon close inspection, it is evident that the refined disparity maps are much smoother both spatially and temporally. Additionally, object edges are still well preserved.


(a) Left View Frame 1 (b) Left View Frame 2 (c) Left View Frame 3

(d) Initial Disparity Frame 1 (e) Initial Disparity Frame 2 (f) Initial Disparity Frame 3

(g) Refined Disparity Frame 1 (h) Refined Disparity Frame 2 (i) Refined Disparity Frame 3

Figure 8.7: Results for surgical video. Notice that the disparity maps refined with the spatio-temporal method are much smoother spatially and temporally.

8.2 Real-time Evaluation

The important thing to consider for the real-time solution is that we do not want to sacrifice algorithmic performance in trade for creating a speed-up. Fig. 8.8 shows the results for the "Book" and "Temple" sequences we evaluated before. Notice that visually the results look even better than our previous methods, implying that the synthesized multiview will also look much better.

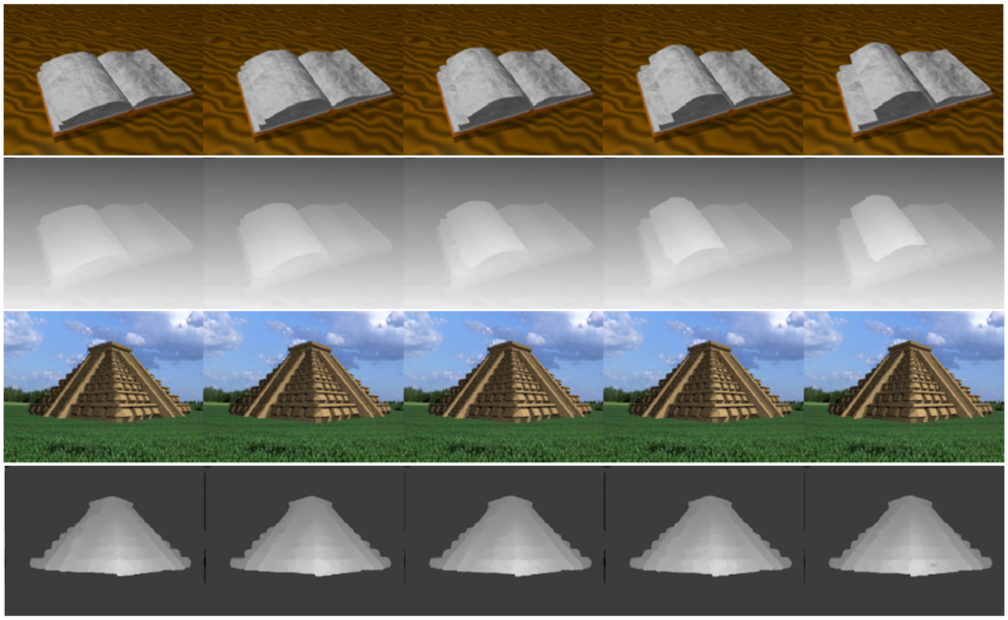
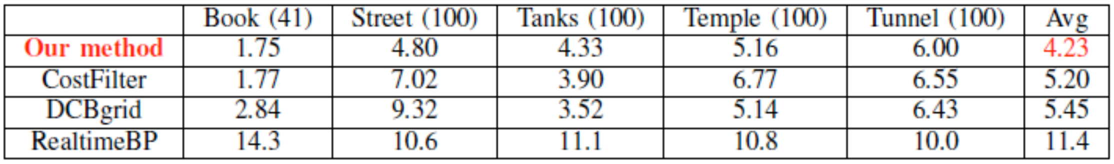


Figure 8.8: Results for real-time disparity estimation method on "Book" and "Temple" sequences.

Table 8.2 quantitatively shows the superiority of our method. We outperform all other real-time or near real-time methods on the four synthetic sequences provided by [35].

Table 8.2: Comparing our real-time method with the top (near) real-time methods.



	Book (41)	Street (100)	Tanks (100)	Temple (100)	Tunnel (100)	Avg
Our method	1.75	4.80	4.33	5.16	6.00	4.23
CostFilter	1.77	7.02	3.90	6.77	6.55	5.20
DCBgrid	2.84	9.32	3.52	5.14	6.43	5.45
RealtimeBP	14.3	10.6	11.1	10.8	10.0	11.4

8.3 View Synthesis Evaluation

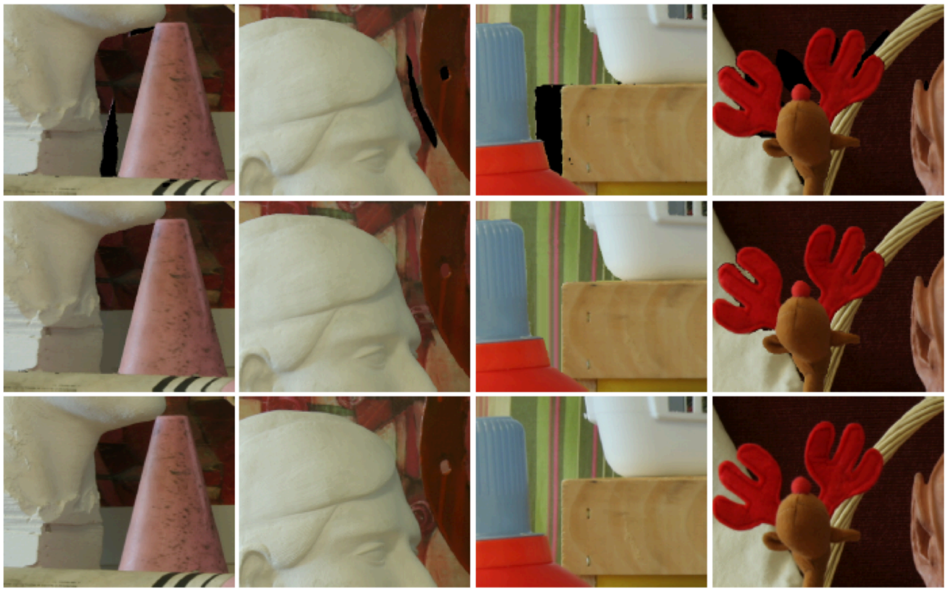


Figure 8.9: Synthesized views using the coordinate alignment approach. Top row: views with holes still unfilled. Middle row: Result of the hole filling. Bottom row: Ground truth.

Analysis of the performance of our algorithms is an important element in gauging the quality of the 3D video, particularly in how well we synthesize the intermediate views. To evaluate the performance of our methods we make use of the Middlebury datasets [25, 79], widely referenced in the area of stereo correspondence. The datasets are used to evaluate state-of-the-art correspondence algorithms, but we take advantage of the fact that they provide multiple views with respective ground truth disparity maps. Each of their datasets contains 7 views with true disparity maps for views 1 and 5.

For each dataset, we use views 1 and 5 and their disparity maps to generate synthetic images for views 2, 3, and 4 using our methods. Then for each of the three views from the dataset, we compute the peak signal-to-noise-ratio (PSNR) between the novel image and the original image found in the database.

While PSNR is a good measure of an algorithm's ability to reconstruct the original image, the human visual system does not actually perceive PSNR. In fact, an image could have a relatively low PSNR value (<25dB) and still appear acceptable to the human eye while the converse is not true. Consequently, to further evaluate our algorithms' abilities to reconstruct a new view, we examine the mean structural similarity (SSIM) index [87] between the ground truth and the synthesized views. If we regard the original image as perfect, then the SSIM is a quality measure of the synthesized image, with a range of $[0, 1]$ and a score of 1 for a perfect match.

Table 8.3: Coordinate Alignment Approach

PSNR and SSIM Averaged over 3 Views				
	Proposed		Tran *et al.* [75]	
Dataset	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Art	32.82	0.95	32.66	0.95
Books	31.06	0.95	30.92	0.93
Cloth1	35.99	0.97	35.99	0.97
Dolls	33.22	0.95	33.05	0.95
Laundry	32.16	0.95	32.13	0.95
Moebius	34.58	0.96	34.30	0.94
Monopoly	32.27	0.95	32.19	0.95
Plastic	37.90	0.98	37.77	0.98
Reindeer	34.10	0.95	33.70	0.94
Wood	37.50	0.94	37.47	0.94

Table 8.4: Efficient Stereo-to-Multiview Approach

PSNR and SSIM Averaged over 3 Views				
	Proposed		Tran *et al.* [75]	
Dataset	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Art	31.67	0.95	32.66	0.95
Books	30.10	0.93	30.92	0.93
Cloth1	35.04	0.96	35.99	0.97
Dolls	31.61	0.95	33.05	0.95
Laundry	31.66	0.95	32.13	0.95
Moebius	33.42	0.95	34.30	0.94
Monopoly	29.80	0.95	32.19	0.95
Plastic	37.91	0.98	37.77	0.98
Reindeer	32.79	0.95	33.70	0.94
Wood	36.29	0.94	37.47	0.94

Fig. 8.9 shows some synthesized views using the coordinate alignment method. They images reveal that aesthetically we are able to generate visually-

pleasing novel views. To quantitatively evaluate the performance, we use PSNR and SSIM. Table 8.3 has results for views 2, 3, and 4 averaged for each dataset using our coordinate alignment method and Table 8.4 provides the results for our efficient approach. Both methods are compared with the state-of-the-art method of [75], showing that we get excellent results.

Lastly, Fig. 8.10 shows the synthesis of six virtual views given two original left and right stereo laparoscopic views. Visually, the results look more than adequate, but we are unable to quantitatively evaluate the performance since no ground truth data exists in the surgical environment. However, in the next chapter, we will discuss a preliminary study we performed with surgical fellows using our system that yielded extremely promising results.

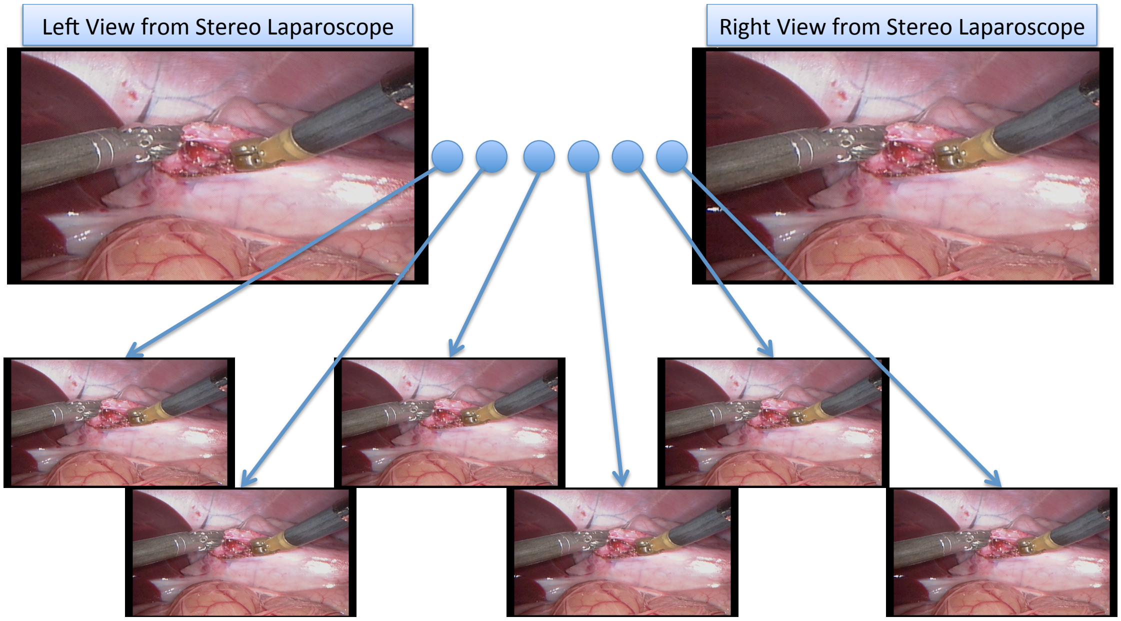

Left View from Stereo Laparoscope
Right View from Stereo Laparoscope

Figure 8.10: Synthesizing the six virtual views to enable glasses-free visualization on the autostereoscopic display in the operating room.

Chapters 4, 5, and 8, in part, are reprints of the material as it appears in [1, 2, 3]. The dissertation author was the primary investigator and author in [1] and secondary in [2, 3].

Chapters 6 and 8, in part, are reprints of the material as it appears in [4, 5]. The dissertation author was a secondary investigator and author in these works.

Chapter 9

Validation Study

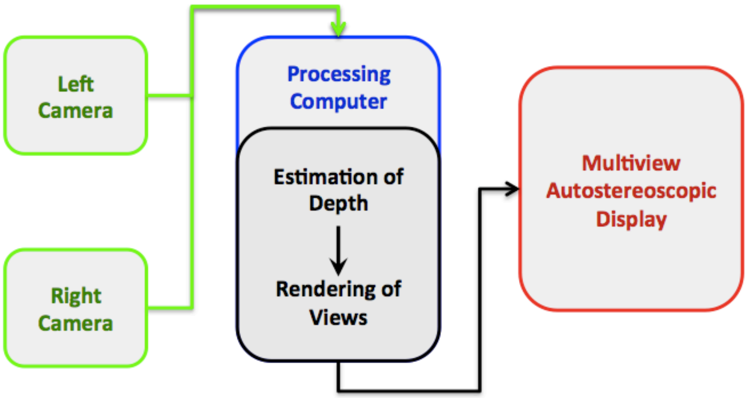


Figure 9.1: Flowchart of how the stereoscopic footage coming into the laparoscope ends up being visualized in multiview 3D.

9.1 System Overview

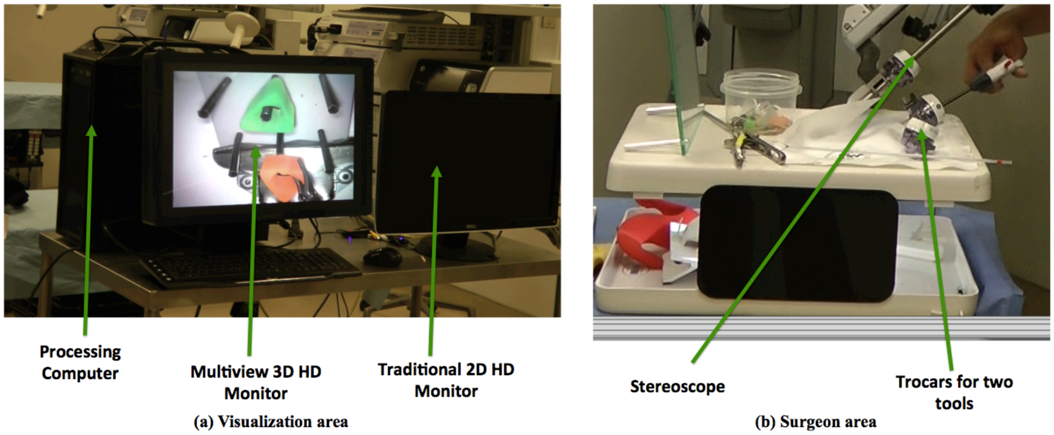
Fig. 9.1 depicts a high-level schematic flowchart of how the stereo video feeds proceed from the operative field all the way to the viewer, where they are visualized in 3D on the multiview display. We use a 10-mm dual-channel laparoscope with stereo endoscopic cameras. The laparoscope captures a stereoscopic pair of video feeds of the scene just like a conventional 2D scope would. This footage is then delivered to the processing computer. Here is where our unique, real-time stereo-to-multiview conversion takes the stereoscopic input and processes it for display. This step is necessary in order to accurately display 3D on autostereoscopic monitors and to render a multiview effect. The particular monitor we use requires eight stereoscopic viewing perspectives, but the real advantage is that, with slight modification, our system will function with any multiview display. In short, the system takes the incoming laparoscopic footage, estimates depth in the scene, and renders the additional perspectives to correctly display multiview 3D. All of this is done in real time so that camera acquisition is the only latency in the system.

9.2 Experimental Design

In our previous work, we thoroughly evaluated the individual components of the proposed system and showed that we achieve state-of-the-art results in terms of standard error metrics on a number of well-known datasets and in terms of generalizability to natural video sequences. To examine the viability of the combined system as a surgical tool, however, we chose to perform an initial user study with some common laparoscopic tasks.

Three laparoscopically experienced surgeons performed multiple trials of two tasks with both a standard 2D laparoscopic system and the new 3D system. Performance was evaluated on accuracy based on a predefined set of errors and also on task completion time. Additionally, a subjective assessment was carried out to gauge elements such as depth perception, visual comfort level, and overall quality.

Fig. 9.2 details the various components of the system as used in this experiment. In Fig. 9.2(a), the visualization components are depicted. To minimize variation in the comparison between 3D and 2D viewing, a standard 2D HD monitor (1080p at 60Hz) was placed adjacent to the multiview 3D monitor (also 1080p at 60Hz) from Alioscopy, Inc. [23]. The processing computer is shown in the background. The stereo camera feeds are directed into this machine, allowing us to display either 2D or multiview 3D at will, based on which monitor is being utilized at the moment. Fig. 9.2(b) shows the surgeon area, located roughly 2 meters from



(a) Visualization area

(b) Surgeon area

Figure 9.2: Complete experimental setup including stereoscope, 2D and 3D monitors, and the processing computer.

the monitors. The operative field is concealed from the participant's eyes and the tools must be inserted through a pair of trocars as in standard laparoscopic surgeries. The scope is angled so that it is looking downward onto the operative field in the same orientation as the participant.

9.2.1 Task 1 - Peg Transfer

The purpose of the first task was to evaluate the efficacy of glasses-free 3D viewing. In particular, our goal was to see if there was any added benefit to incorporating the element of depth perception and if it had any significant bearing on laparoscopically experienced surgeons.

Therefore, we chose a peg transfer task, which naturally has a large amount of depth. In Fig. 9.3, the individual steps of the task are outlined with white arrows. The participant's objective was to relocate the green and orange objects from the left-most pegs all the way to the right using a single grasper in the particular order depicted in the figure.

To evaluate the participants, we noted the number of attempts each transfer took. A missed grasp or false placement added to a tally for each destination peg. Also, task completion times were recorded for each trial.

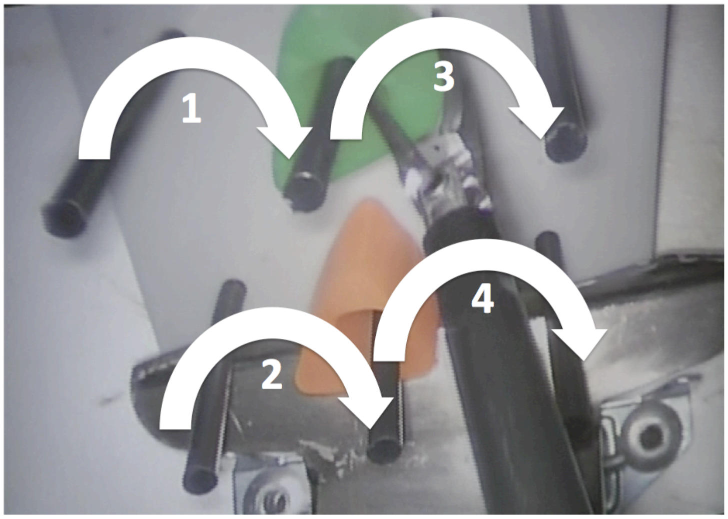


Figure 9.3: The peg transfer task. Participants had to move the objects consecutively from peg to peg in the numbered order.

9.2.2 Task 2 - Suturing

One of the many advantages of our system is the ability we have to control the degree of depth an individual perceives. From the stereoscopic footage, we can render the multiview 3D to provide absolutely no depth (analogous to a 2D image) or full 3D, as would be seen with conventional glasses-based systems. We can even extrapolate beyond the standard camera baseline to provide an even more immersive 3D experience.

In the cinema world, it is often the case that too much depth leads to headaches, nausea, or discomfort. This has often been attributed to the vergence-accommodation conflict [88], since the eyes are focused on the screen but are verging towards differing depth planes. This unnatural behavior is thought to be the primary source of most negative 3D experiences. Without having to wear glasses (as in our system), however, the eyes are free to saccade towards physical objects outside of the display boundaries, potentially reducing much of the normal discomfort.

In view of this, the aim of the second task was to gauge exactly how much depth we could make the participants experience without it becoming uncomfortable or even detrimental to the task. We asked participants to perform a simple

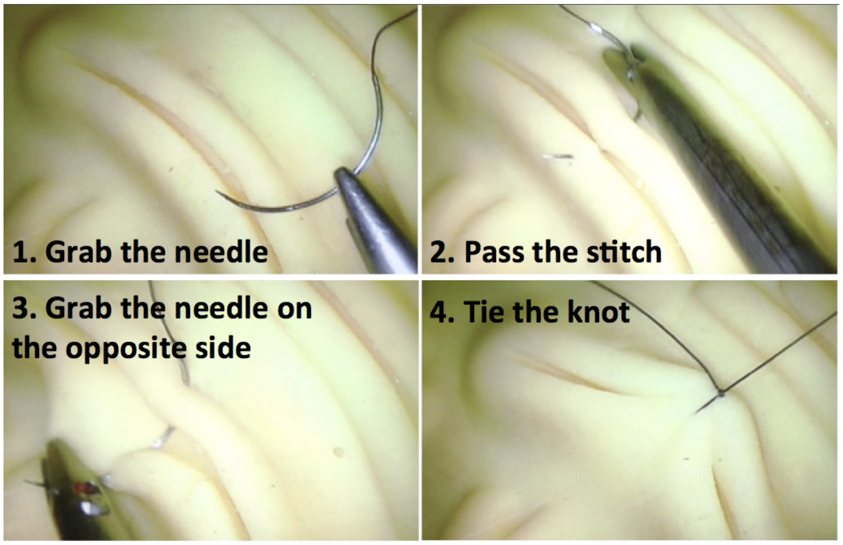


Figure 9.4: The suturing task. Participants were asked to perform the steps in order as quickly and accurately as possible.

suturing task. The needle was inserted into the left trocar and grabbed with the right hand once visible, the stitch was passed from right to left, the needle was grabbed on the opposite side, and finally a single loop knot was tied. Fig. 9.4 illustrates these four steps.

The task was performed twice, both times with the 3D system. The first trial was performed with minimal 3D. The second trial contained the normal amount of depth. As in the Peg Transfer task, every move in each of the four steps in this task was logged. Any form of misrepresentation (drop, miss, etc.) was tallied as an additional attempt. In the end, the total task completion time was recorded as well.

9.3 Results

Tables 9.1 and 9.2 show the average results for each task among the study participants. In Table 9.1, we see that, while the task completion time was greater for the first trial with multiview 3D, it dropped significantly in the second trial. Additionally, fewer errors were made with 3D than 2D. Table 9.2 shows the results for the suturing task. While the number of errors was comparable, the task

Table 9.1: Peg Transfer Task Results Averaged Over All Participants

Trial Number	**Traditional 2D Laparoscopy**	
	Completion Time (s)	*Number of Errors*
1	20.64	2.00
2	21.26	2.33

Trial Number	**Multiview 3D Laparoscopy**	
	Completion Time (s)	*Number of Errors*
1	23.30	1.33
2	17.49	1.33

Table 9.2: Suturing Task Results Averaged Over All Participants

Trial Number	**Minimal 3D (Low depth resolution)**	
	Completion Time (s)	*Number of Errors*
1	60.34	0.50
2	57.25	0.50

Trial Number	**Full 3D (High depth resolution)**	
	Completion Time (s)	*Number of Errors*
1	41.46	1.00
2	39.66	0.00

completion time was significantly lower with greater depth perception.

For the peg transfer task, completion times were generally comparable. However, as the results show, 3D times improved significantly between trials. Considering that the subjects were all experienced with traditional laparoscopy, this finding reveals that using the system is quite intuitive and can even overcome years of biased training against it. Also, while slight, the error rates did drop when using the 3D system.

For the longer suturing task, the results were more compelling. While error rates were fairly low and comparable between minimal and full 3D, the completion times were much faster with greater depth. We can see the correlation that as we improve a surgeon's perception of depth, we can expect to see the task becoming easier and more natural. A full perception of 3D is in fact helpful to getting the job done faster without loss in accuracy.

In addition to the quantitative results, a survey was taken to gauge the 2D versus 3D overall experience, the level of difficulty for each task, and the levels of discomfort that were experienced. While the participants had on average about 5 years of laparoscopic experience, making them biased to the 2D monitor, they consistently rated the 3D visual experience higher and agreed that the added depth made their tasks easier to perform. Furthermore, they specifically requested that the amount of depth resolution be further increased. There were no reports of nausea, eye strain, or any form of discomfort. Two of the participants noted it was a little strange for them to work with the multiview display at first because they have been so used to working with a 2D monitor. However, they both quickly acclimated.

The subjective feedback was highly informative. While the numbers in the error rates were not staggeringly different between the two tasks, the surgeons did report that they found it easier to perform them with 3D. In fact, they wanted even greater depth perception than what we were offering them in the study. If laparoscopically experienced surgeons can exhibit that great of an eagerness to work with this new technology, it presents a hopeful case for novice surgeons that have no preferential bias.

Chapter 10

Conclusion

The state-of-the-art in 3D technology is currently in glasses-based systems, whether active or passive. What sets us apart is that no one has ever attempted to deliver 3D to surgeries without the need for glasses before. Of all the situations where depth perception would be necessitated, surgery is clearly one where it is vital since lives are at stake. However, until now, the technology did not exist to develop a feasible system.

Our methods for disparity estimation and view synthesis are some of the fastest and most accurate in the world to date. These advanced algorithms, combined with highly optimized GPU programming, present us with the unique opportunity to make real-time multiview visualization a reality in the laparoscopic domain. While there is much work ahead of us before we can envision our system being used routinely in surgeries, the research areas are exciting fields and rich with many opportunities, even beyond that of surgery.

We have presented methods by which we may present glasses-free 3D content to surgeons and staff in the OR. We have taken advantage of recent advances in autostereoscopic displays that have enabled them to produce high-quality multiview 3D. Beginning with a stereo sequence, we have shown a robust method by which we may estimate spatio-temporally consistent depth estimates. These depth estimates are then employed in our stereo-to-multiview virtual view synthesis techniques in order to enable us to generate the sequences necessary to show 3D on autostereoscopic displays. We have shown results that illustrate the robustness of our techniques on data from varied sources with little to no adjustment of parameters. Additionally, the validation study we performed reaffirmed the fact that this system could be an extremely useful tool in laparoscopic surgeries. Not only did the surgeons acclimate quickly to the multiview perception, they also demonstrated a strong preference for the 3D experience, even though their years of training in 2D laparoscopy clearly presented a bias for them.

Although our system is designed as a supplement to OR technologies, such as the DaVinci, it could be a cost-effective alternative for institutions unable to pay millions of dollars for a full surgical robot. This system would decrease the amount of equipment necessary for surgery and the cost would be driven down tremendously.

Our multiview glasses-free 3D surgical prototype is the first of its kind and still requires further evaluation. However, the results of our preliminary validation study are extremely promising and show that the system does in fact aid in surgical efficacy and is a preferred viewing method over traditional 2D monitors.

We intend to further evaluate this prototype with a wider range of parameters. We plan to perform a full study using medical students with no laparoscopic experience, experienced surgeons, and expert surgeons that have over 10 years of experience. We intend to add additional experimental tasks to assess not just task completion accuracy but accuracy in localizing small details, such as blood vessels, in three-dimensional space. We hope that our findings will be consistent with our expectation that our system could dramatically improve the practice of laparoscopy and one day perhaps actually be utilized.

Bibliography

[1] R. Khoshabeh, S. H. Chan, and T. Q. Nguyen, "Spatio-temporal Consistency in Video Disparity Estimation," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 885–888, 2011.

[2] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An Augmented Lagrangian Method for Video Restoration," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 941–944, 2011.

[3] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Video Restoration," *Transactions on Image Processing*, vol. 20, pp. 3097–3111, January 2011.

[4] L. Tran, R. Khoshabeh, A. K. Jain, C. Pal, and T. Q. Nguyen, "Spatially Consistent View Synthesis with Coordinate Alignment," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[5] A. K. Jain, L. Tran, R. Khoshabeh, and T. Q. Nguyen, "Efficient Stereo-to-Multiview Synthesis," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[6] S. H. Kong, B. M. Oh, H. Yoon, H. S. Ahn, H. J. Lee, S. G. Chung, N. Shiraishi, S. Kitano, and H. K. Yang, "Comparison of Two- and Three-dimensional Camera Systems in Laparoscopic Performance: A Novel 3D System with One Camera," *Surgical Endoscopy*, vol. 24, pp. 1132–1143, 2010.

[7] I. Jourdan, E. Dutson, A. Garcia, T. Vleugels, J. Leroy, D. Mutter, and J. Marescaux, "Stereoscopic Vision Provides a Significant Advantage for Precision Robotic Laparoscopy," *British Journal of Surgery*, vol. 91, pp. 879–885, 2004.

[8] H. Tevaearai, X. Mueller, and L. von Segesser, "3D Vision Improves Performance in a Pelvic Trainer," *Endoscopy*, vol. 32, 2000.

[9] J. Byrn, S. Schluender, C. Divino, J. Conrad, B. Gurland, E. Shlasko, and A. Szold, "Three-dimensional Imaging Improves Surgical Performance for

Both Novice and Experienced Operators using the Da Vinci Robot System," *American Journal of Surgery*, vol. 193, pp. 519–522, 2007.

[10] K. Votanopoulos, F. Brunicardi, J. Thornby, and C. Bellows, "Impact of Three-dimensional Vision in Laparoscopic Training," *World Journal of Surgery*, vol. 32, pp. 110–118, 2008.

[11] H. R. Patel, M. J. Ribal, M. Arya, R. Nauth-Misir, and J. Joseph, "Is It Worth Revisiting Laparoscopic Three-dimensional Visualization? A Validated Assessment," *Urology*, vol. 70, pp. 47–49, 2007.

[12] L. Way, L. Stewart, W. Gantert, K. Liu, C. M. Lee, K. Whang, and J. Hunter, "Causes and Prevention of Laparoscopic Bile Duct Injuries," *Annals of Surgery*, vol. 237, no. 4, pp. 460–469, 2003.

[13] R. Berguer, W. D. Smith, , and Y. H. Chung, "Performing Laparoscopic Surgery is Significantly More Stressful for the Surgeon than Open Surgery," *Surgical Endoscopy*, vol. 15, pp. 1204–1207, 2001.

[14] "Intuitive Surgical Inc." http://www.intuitivesurgical.com.

[15] V. Ramachandran, "3D Fundamentals," tech. rep., University of California, San Diego, 2009.

[16] N. A. Valyus, "Stereoscopy," in *The Focal Press*, 1966.

[17] T. Okoshi, "Three-dimensional Imaging Techniques," in *Academic Press*, 1976.

[18] B. Lane, "Stereoscopic Displays," in *Society of Photographic Instrumentation Engineers*, vol. 367, 1982.

[19] D. F. McAllister, "Stereo Computer Graphics and Other True 3D Technologies," in *Princeton University Press*, 1993.

[20] I. Susumu, T. Nobuji, and I. Morito, "Technique of Stereoscopic Image Display," *EP at. No 0 354 851*, June 1990 (filed in Japan in August 1988).

[21] N. A. Dodgson, "Multi-view Autostereoscopic 3D Display," *Stanford Workshop on 3D Imaging*, 2011.

[22] A. Redert, R. P. Berretty, C. Varekamp, O. Willemsen, J. Swillens, and H. Driessen, "Philips 3D Solutions: From Content Creation to Visualization," in *3D Data Processing, Visualization, and Transmission*, 2006.

[23] "Alioscopy Inc." http://www.alioscopy.com.

[24] P. An, Z. Zhang, and L. Shi, "Theory and Experiment Analysis of Disparity for Stereoscopic Image Pairs," in *International Symposium on Intelligent Multimedia, Video, and Speech Processing*, pp. 68–71, 2001.

[25] H. Hirschmuller and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," in *CVPR*, 2007.

[26] D. Scharstein and C. Pal, "Middlebury Stereo Evaluation Dataset Website." http://vision.middlebury.edu/stereo.

[27] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *PAMI*, vol. 23, pp. 1222–1239, February 2004.

[28] V. Kolmogorov and R. Zabih, "Computing Visual Correspondence with Occlusions via Graph Cuts," in *ICCV*, pp. 508–515, 2001.

[29] P. Felzenszwalb and D. Huttenlocher, "Efficient Belief Propagation for Early Vision," in *CVPR*, pp. 261–268, 2004.

[30] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 492–504, March 2009.

[31] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *IJCV*, vol. 47, pp. 7–42, April 2002.

[32] O. Williams, M. Isard, and J. MacCormick, "Estimating Disparity and Occlusions in Stereo Video Sequences," in *CVPR*, 2005.

[33] F. Huguet and F. Devernay, "A Variational Method for Scene Flow Estimation from Stereo Sequences," in *ICCV*, 2007.

[34] M. Bleyer and M. Gelautz, "Temporally Consistent Disparity Maps from Uncalibrated Stereo Videos," in *ISPA*, 2009.

[35] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson, "Real-time Spatiotemporal Stereo Matching Using the Dual-Cross-Bilateral Grid," in *ECCV*, 2010.

[36] K. J. Yoon and I. S. Kweon, "Locally Adaptive Support-Weight Approach for Visual Correspondence Search," in *CVPR*, 2005.

[37] J. Zhu, L. Wang, R. Yang, and J. Davis, "Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps," in *CVPR*, pp. 1–8, 2008.

[38] G. Zhang, J. Jia, T. T. Wong, and H. Bao, "Consistent Depth Maps Recovery from a Video Sequence," *PAMI*, vol. 31, no. 6, pp. 974–988, 2009.

[39] R. Gonzales and R. Woods, *Digital Image Processing.* Prentice Hall, 2007.

[40] L. Lucy, "An Iterative Technique for the Rectification of Observed Distributions," *Astronomical Journal*, vol. 79, p. 745, 1974.

[41] W. Richardson, "Bayesian-based Iterative Method of Image Registration," *JOSA-A*, vol. 62, no. 1, pp. 55–59, 1972.

[42] V. Mesarovic, N. Galatsanos, and A. Katsaggelos, "Regularized Constrained Total Least-Squares Image Restoration," *IEEE Transactions on Image Processing*, vol. 4, pp. 1096–1108, 1995.

[43] P. Hansen, J. Nagy, and D. O'Leary, "Deblurring Images: Matrices, Spectra, and Filtering," *Fundamentals of Algorithms 3*, 2006.

[44] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[45] T. Chan, G. Golub, and P. Mulet, "A Nonlinear Primal-Dual Method for Total Variation-based Image Restoration," *SIAM Journal on Scientific Computing*, vol. 20, pp. 1964–1977, 1999.

[46] A. Chambolle, "An Algorithm for Total Variation Minimization and Applications," *Journal of Math, Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[47] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast Image Recovery using Variable Splitting and Constrained Optimization," *IEEE Transactions on Image Processing*, vol. 19, pp. 2345–2356, September 2010.

[48] J. Yang, W. Yin, and Y. Zhang, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM Journal on Scientific Computing*, vol. 31, pp. 2842–2865, 2009.

[49] Y. Wang, J. Osterman, and Y. Zhang, *Video Processing and Communications.* Prentice Hall, 2002.

[50] B. Lucas, *Generalized Image Matching by the Method of Differences.* PhD thesis, Carnegie Mellon University, 1984.

[51] Q. Shan, J. Jia, and A. Agarwala, "High-quality Motion Deblurring from a Single Image," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 27, no. 3, 2008.

[52] S. Dai and Y. Wu, "Motion from Blur," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2008.

[53] A. Levin, "Blind Motion Deblurring using Image Statistics," in *Neural Information Processing Systems (NIPS)*, 2006.

[54] S. Cho, Y. Matsushita, and S. Lee, "Removing Non-uniform Motion Blur from Images," in *International Conference on Computer Vision (ICCV)*, 2007.

[55] M. Ng, H. Shen, E. Lam, and L. Zhang, "A Total Variation Regularization based Super-Resolution Reconstruction Algorithm for Digital Video," *EURASIP Journal on Advances in Signal Processing*, 2007.

[56] S. Belekos, N. Galatsanos, and A. Katsaggelos, "Maximum A Posteriori Video Super-Resolution using a New Multichannel Image Prior," *IEEE Transactions on Image Processing*, vol. 19, pp. 1451–1464, 2010.

[57] S. Chan and T. Q. Nguyen, "LCD Motion Blur: Modeling, Analysis, and Algorithm," *IEEE Transactions on Image Processing*, 2011.

[58] B. Jahne, "Spatio-temporal Image Processing: Theory and Scientific Applications," *Springer*, 1993.

[59] Y. Wexler, E. Shechtman, and M. Irani, "Space-time Completion of Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, pp. 1–14, 2007.

[60] E. Shechtman, Y. Caspi, and M. Irani, "Space-time Super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, pp. 531–545, 2005.

[61] S. Farsiu, M. Elad, and P. Milanfar, "Video-to-Video Dynamic Super-resolution for Grayscale and Color Sequences," *EURASIP Journal on Applied Signal Processing*, 2006.

[62] Y. Huang, M. Ng, and Y. Wen, "A Fast Total Variation Minimization Method for Image Restoration," *SIAM Multiscale Model and Simulation*, vol. 7, pp. 774–795, 2008.

[63] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[64] J. Douglas and H. Rachford, "On the Numerical Solution of Heat Conduction Problems in Two and Three Space Variables," *Transactions of the American Mathematical Society*, vol. 82, pp. 421–439, 1956.

[65] R. Rockafellar, "Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming," *Mathematics of Operations Research*, vol. 1, pp. 97–116, 1976.

[66] R. Rockafellar, "Monotone Operators and Proximal Point Algorithm," *SIAM Journal on Control and Optimization*, vol. 14, 1976.

[67] S. H. Chan, P. E. Gill, and T. Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Image Restoration," *IEEE Transactions on Image Processing*, June 2010.

[68] D. Bertsekas, "Multiplier Methods: A Survey," *Automatica*, vol. 12, pp. 133–145, 1976.

[69] S. M. Seitz and C. R. Dyer, "View Morphing," in *SIGGRAPH*, 1996.

[70] I. Gorodnitsky and B. Rao, "Sparse Signal Reconstruction from Limited Data using FOCUSS: A Re-weighted Minimum Norm Algorithm," in *IEEE Transactions on Signal Processing*, 1997.

[71] K. Oh, S. Yea, and Y. Ho, "Hole-Filling Method Using Depth Based In-Painting for View Synthesis in Free Viewpoint Television (FTV) and 3D Video," in *Mitsubishi Electric Research Laboratories*, TR2009-25, 2009.

[72] C. Cheng, S. Lin, S. Lai, and J. Yang, "Improved Novel View Synthesis from Depth Image with Large Baseline," in *ICPR*, pp. 1–4, 2008.

[73] Y. Fan and T. Chi, "The Novel Non-Hole-Filling Approach of Depth Image Based Rendering," in *3DTV-CON*, 2008.

[74] C. Fehn, "Depth-image-based Rendering, Compression, and Transmission for a New Approach on 3D-TV," *SPIE Stereoscopic Displays and Virtual Reality Systems XI*, 2004.

[75] L. Tran and T. Nguyen, "View Synthesis Based on Conditional Random Fields and Graph-cuts," in *ICIP*, 2010.

[76] S. J. Lin, C. M. Cheng, and S. H. Lai, "Spatio-temporally Consistent Multi-view Video Synthesis for Autostereoscopic Displays," *PCM*, vol. 5879, pp. 532–542, 2009.

[77] A. Criminisi, P. Prez, and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *TIP*, vol. 13, pp. 1200–1212, 2004.

[78] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *PAMI*, vol. 24, pp. 603–619, 2002.

[79] D. Scharstein and C. Pal, "Learning Conditional Random Fields for Stereo," in *CVPR*, ', 2007.

[80] S. C. Chan, H. Y. Shum, and K. T. Ng, "Image-based Rendering and Synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 6, 2007.

[81] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[82] "CUDA by NVIDIA." http://www.nvidia.com.

[83] R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," *Springer-Verlag*, pp. 151–158, 1994.

[84] S. Birchfield and C. Tomasi, "A Pixel Dissimilarity Measure that is Insensitive to Image Sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 20, pp. 401–406, April 1998.

[85] K. Zhang, J. Lu, and G. Lafruit, "Cross-based Local Stereo Matching using Orthogonal Integral Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, 2009.

[86] Mobile 3DTV. http://sp.cs.tut.fi/mobile3dtv/stereo-video/.

[87] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Measurement to Structural Similarity," in *IEEE Transactions on Image Processing*, vol. 13, 2004.

[88] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks, "Vergence-accomodation conflicts hinder visual performance and cause visual fatigue," *Vision*, vol. 8, pp. 1–30, 2008.