OCCLUSION-AWARE INTERMEDIATE VIEW RECONSTRUCTION

SERDAR İNCE

Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

BOSTON UNIVERSITY

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

OCCLUSION-AWARE INTERMEDIATE VIEW RECONSTRUCTION

by

SERDAR İNCE

B.S., Middle East Technical University, 2000
M.S., Middle East Technical University, 2002

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2008

Approved by

First Reader

Janusz Konrad, Ph.D.
Professor of Electrical and Computer Engineering

Second Reader

W. Clem Karl, Ph.D.
Professor of Electrical and Computer Engineering

Third Reader

Michael Gennert, Ph.D.
Professor of Computer Science
Worcester Polytechnic Institute

Fourth Reader

Hanspeter Pfister, Ph.D.
Gordon McKay Professor of the Practice of Computer Science
Harvard University

To my beloved wife Gözde;

who always supports me...

Acknowledgments

Many wonderful people helped me during my PhD. I want to thank them with a couple of words, although a few words would never be enough to show my gratitude.

First and foremost, I want to express my sincere gratitude to my advisor Prof. Janusz Konrad. Without his guidance, advice, and mentorship, this thesis would not be possible at all. His endless patience and full trust helped me to overcome the difficulties of research and being a graduate student. He enthusiastically shared his broad knowledge and suggested ways to improve myself. The professional skills that I learned from him will help me all through my career. In addition to being an excellent mentor, he is also a great person to talk about anything. I feel very lucky to have met and worked with him.

This thesis would not be complete without the precious feedback of my thesis committee members Prof. Clem Karl, Prof. Michael Gennert and Prof. Hanspeter Pfister.

I would like to thank Prof. Karl for his constructive comments on my work. I would also like to thank him for his image reconstruction class. It was one of the most rewarding classes I have taken at Boston University, which greatly helped me in my research.

Prof. Gennert's valuable advice and suggestions, for which I am very thankful, helped me to improve my work. One of his suggestions greatly helped me in improving the optical flow algorithm in my thesis.

I would like to thank Prof. Pfister for his constructive feedback on my work. Especially his valuable suggestions in my prospectus defense broadened the perspective of my work and helped me to improve my thesis.

During my PhD, I had the opportunity and pleasure to work at Mitsubishi Electric Research Laboratories. I would like to thank my supervisor Dr. Anthony Vetro for inviting me to MERL. I would also like to thank him, Dr. Emin Martinian and Dr. Sehoon Yea for helping me during my internship.

I would like to thank Prof. Lars Oddsson of NeuroMuscular Research Center of Boston University for our collaboration and his help in our joint project.

I would to thank Dr. Carlos Vazquez of Communication Research Centre Canada for our discussion on splines.

I would like to thank our former group members Mirko Ristivojevic, Nikola Bozinovic, Ashish Jain, Mike McHugh, PJ McNerney and Philippe Agniel for their help and friendship during my studies. Moreover, I would like to thank members of Information Systems and Sciences Laboratory and ECE Department Suchin Aeron, Onur Savas, Gorkem Eraslan, Ehsan Afkhami, George Atia, Julia Pavlovich, Birant Orten, Yonggang Shi, Andrew Litvin, Erhan Ermis, Huseyin Mutlu, Ashraf Al Daoud for their friendship and our technical (and usually not-so-technical) discussions.

Without the support of my parents, Hasan and Yaşar, and my sisters Dilek and Zerrin, I would never be able to overcome the difficulties of my life. They endured my absence for all these years and supported me all the time. I cannot thank them enough.

Finally, I would like to thank my dear wife Gözde. Without her endless support, understanding and help, I would never be able to finish this work. There were times that I was about to give up, but she was always there for me and encouraged me to continue. I feel blessed to have her in my life and always by my side.

OCCLUSION-AWARE INTERMEDIATE VIEW RECONSTRUCTION

(Order No.)

SERDAR İNCE

Boston University, College of Engineering, 2008

Major Professor: Janusz Konrad, Ph.D., Professor of Electrical and Computer Engineering

ABSTRACT

This dissertation concentrates on the problem of intermediate view reconstruction, which is defined as follows: given few images of a scene captured by real cameras, reconstruct images that would have been captured by virtual cameras. Our main goal in this dissertation is to reconstruct intermediate views using two input images with a special focus on occlusion areas. Occlusion areas are the areas that are visible only in one of the input images.

We start the dissertation by identifying the main challenges in intermediate view reconstruction from two images, and then offer novel solutions to each challenge. First, we present in detail the popular pivoting-based view reconstruction that requires estimation of a separate disparity field for each view under reconstruction. After pointing out the deficiencies of pivoting-based approach, as an alternative, we propose a new intermediate view reconstruction method based on B-spline approximation. The new approach permits reconstruction of multiple views from a single disparity field, a clear computational advantage. It also assures better robustness to image noise, although is more sensitive to disparity estimation errors than the pivoting-based method. However, most importantly, spline-based reconstruction allows selective forward compensation of visible areas and, therefore, is of importance in occlusion awareness. Next, we present a new simple occlusion area estimation method and show its superior performance over other low-complexity algorithms. The knowledge where occlusions occur in an image is a valuable piece of information since

disparity cannot be reliably estimated there and needs to be inferred in a different manner. In view of this, we present a novel approach to disparity recovery in occlusion areas, namely the image-driven disparity inpainting. We further embed this idea into a variational formulation, and propose occlusion-aware optical flow (disparity) estimation that jointly computes disparity vectors, implicitly detects occlusions and extrapolates disparities in occlusion areas. Combining all of these proposed methods in view reconstruction, we reconstruct realistic and improved intermediate views especially in occlusion areas. Finally, we focus on using multiple images, instead of two images, in view reconstruction to improve the pivoting-based approach. Specifically, we propose another occlusion-aware pivoting-based disparity estimation formulation, which adaptively estimates disparity by using different pairs of input images. The reconstruction using multiple images shows significant improvements over pivoting-based reconstruction that uses two images only.

Intermediate view reconstruction has many applications, especially in the area of 3D displays and communication. One of the most important applications is that it can be used to reconstruct additional views from stereo pairs, so that any stereo pair can be displayed on emerging automultiscopic displays that require many views of a scene as input. It is in this context that we demonstrate applications of the proposed methods in different areas. First, in a biomedical application, the proposed view reconstruction algorithm is embedded into a neuromuscular training system that utilizes automultiscopic 3D displays. Second, the proposed view reconstruction method is applied to monoscopic video sequences to increase frame rate and, therefore, enhance low frame-rate videos captured by mobile phones. Third, the proposed occlusion-aware optical flow method is used to solve a real-world problem of NASA, namely the recovery of a missing color component in stereo images. Finally, disparity estimation in a multiview video codec that uses view reconstruction is shown to benefit from our methods.

Contents

List of Tables

List of Figures

List of Abbreviations

1D	.....	One Dimensional
2D	.....	Two Dimensional
3D	.....	Three Dimensional
AVC	.....	Advanced Video Coder
CIE	.....	Commission Internationale de l'Eclairage
CRT	.....	Cathode Ray Tube
DWT	.....	Discrete Wavelet Transform
FPS	.....	Frames Per Second
HDTV	.....	High-Definition Television
HMSD	.....	Head Mounted Stereo Display
IVR	.....	Intermediate View Reconstruction
LCD	.....	Liquid Crystal Display
MPEG	.....	Moving Pictures Expert Group
NASA	.....	National Aeronautics and Space Administration
NMRC	.....	NeuroMuscular Research Center
NTSC	.....	National Television System Committee
OF	.....	Optical Flow
PSNR	.....	Peak-Signal-to-Noise-Ratio
TV	.....	Television

Chapter 1

Introduction

This dissertation concentrates on the problem of *intermediate view reconstruction*, which is defined as follows: given few images of a scene captured by real cameras, reconstruct images that would have been captured by virtual cameras. The problem is illustrated in Fig. 1·1. We would like to reconstruct an image from the virtual camera by using images captured by cameras #1 and #2.

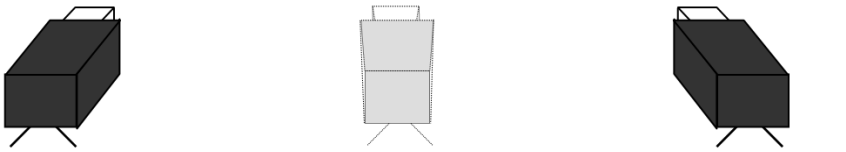


Figure 1·1: Illustration of intermediate view reconstruction; reconstruct virtual camera image from images captured by cameras #1 and #2.

The problem is of interest for it finds various applications, among others in 3D display systems such as the emerging eyewear-free multiview 3D displays. Such displays are designed to project multiple views of a scene that are not always available. Intermediate view reconstruction can be used to generate the additional views by using few captured images, thus enabling multiview displays even when very few cameras are used to capture a scene.

Our focus in intermediate view reconstruction is on the impact of occlusions. Occlusions occur when certain image areas are visible in some images only. For example, a part of the scene may be visible in camera #1 but not visible in camera #2. Occlusion areas pose significant challenges in view reconstruction as will be discussed in this dissertation.

This chapter contains a general introduction to the problem and motivation for this dissertation. First, it discusses the evolution of image and display technologies. This is

followed by an overview of benefits of 3D systems and difficulties involved in their design. Then, intermediate view reconstruction is discussed in more detail along with its applications.

1.1 Evolution of display and image technologies

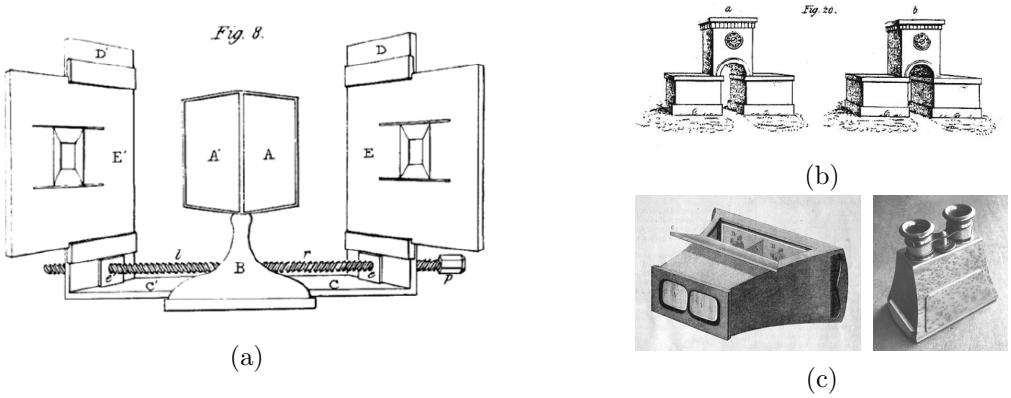
Thanks to the advances in display technology and signal processing, nowadays we have vast opportunities to make our lives easier using visualization tools. Visualization tools find applications in wide range of areas, from entertainment to medicine, from surveillance to communications. They fundamentally changed how we live.

The first black and white still image was captured by Nicéphore Niépce in 1816. Early black and white photos achieved high quality over the years, and were later enhanced by the addition of color. To people's fascination, still pictures were followed by moving pictures.

The twentieth century witnessed an impressive use of images. Television is invented by using communication technologies that allowed to transmit pictures. Then, the digital era slowly started to replace analog devices of the imaging technologies. Digital cameras changed the way images are captured. In the last twenty years, the picture quality reached an impressive level. High-definition television, for example, with its 1920×1080 resolution, amazes audiences.

Unfortunately, there is still a missing piece in current imaging technologies: the appreciation of depth. Adding the depth to the state-of-the-art visualization tools will enhance the quality and offer a more realistic experience to the viewer. Flight/battlefield simulators, fields of medicine and entertainment are among those to benefit from this realistic experience.

Humans perceive depth thanks to the coordination of their eyes and brain. Since the eyes are separated by a about 2.5 inches, they perceive the scene from two slightly different angles. These images are then fused by the brain to perceive depth. This is the basic idea behind the stereoscopic vision. Stereoscopic images are captured by two adjacent cameras mimicking the way humans see the world. The basics of stereoscopy and its applications



(a) (b) (c)

Figure 1·2: (a) Diagram of Wheatstone's stereoscope from his paper in 1838. Mirrors A and A' are used to shows different images to each eye (b) Wheatstone used hand drawings to demonstrate the function of his stereoscopic device (c) Brewter-type stereoscopes (from `http://users.telenet.be/thomasweynants/stereoscope.html`).

will be discussed later in this dissertation.

1.1.1 Early stereoscopy

It is actually very interesting that the work on three dimensional images had begun long before photography was invented. It is noted that Leonardo da Vinci (1452-1519) studied the perception of depth. His interest and appreciation for the third dimension can be found in his notes written 500 years ago. He writes *"...the main objective of painting is to show a raised body projecting from a plane surface. Whoever achieves this surpasses all others and should be considered most skilled within his profession."* (Benyon, 1998).

However, the breakthrough happened in 1838 when British physicist Charles Wheatstone (1802-1875) invented the first stereoscopic viewer. Wheatstone explained the theory of stereoscopic vision in his address entitled *"Phenomena of Binocular Vision"* to the Royal Scottish Society of Arts in June 1838 (Wheatstone, 1838). He constructed a stereoscopic viewing device knowing that left and right eyes view the same scene from slightly different angles. His device consisted of two mirrors, shown with A' and A in Fig. 1·2.a (Wheatstone, 1838), and two planes, E and E', to attach the pictures for the left and right eyes. He used hand drawings to demonstrate the device (Fig. 1·2.b).

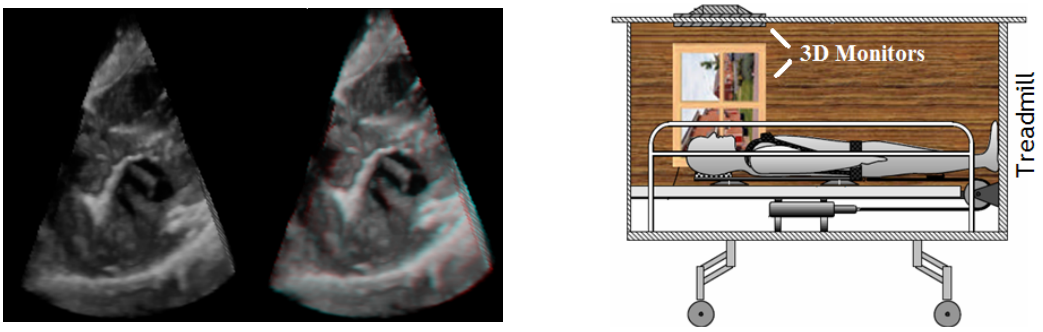
Following Wheatstone, David Brewster (1781-1868) made significant contributions to the field of stereoscopy. In 1849, he proposed his own stereoscope consisting of a pair of half lenses and an opening to a slot where the pair of images could be mounted side by side. This new design, shown in Fig 1·2.c, was more compact than that of Wheatstone's and later became a template for many subsequent stereoscopes. One year later, Brewster succeeded in interesting the French optics company Soleil and Duboscq and the company started manufacturing the device (Howard and Howard, 1995). Soleil and Duboscq exhibited the stereoscope in London. Queen Victoria was very impressed by the device. Having caught the attention of a noble, it easily drew the public's attention. As a result, within a three months period, a quarter of a million stereoscopes were sold in London and Paris.

More sophisticated methods to deliver the left and right images to each eye are proposed in time. Early attempts were anaglyph glasses which are composed of red and cyan colored lenses. Later, new devices such as shutter and polarized glasses were proposed to deliver a better quality. Recent systems even eliminated the need for eyewear by using lenticular sheets or parallax barriers. We will present a detailed analysis of technology for 3D visualization in Section 2.4. Let us first outline the benefits of using 3D visualization in various fields.

1.2 Benefits and applications of stereoscopic display technologies

Many applications can benefit from stereoscopic vision because the depth cues in 2D images and videos (called 'monocular depth cues', to be discussed in Section 2.1.1) are not enough to fully perceive depth. Even worse, they may be misleading; some of the popular optical illusions are due to the incorrectly perceived depth cues. Through stereoscopic vision, we can introduce a realistic depth feeling to the display for the viewer.

Potential applications of stereoscopic vision in the field of medicine are numerous. Standard displays in computer tomography and other radiology equipment can be replaced by stereoscopic systems which can create a more realistic experience and can make diagnosis easier (Hubbold et al., 1997; Wang et al., 2004). As shown in ultrasound scan in



(a) (b)

Figure 1·3: (a) An ultrasound scan (Property of Harvard University Medical School) in 2D (left) and in 3D (right; anaglyph glasses required) (b) an exercise bed for bedridden patients equipped with 3D displays.

Fig. 1·3.a, the 3D version on the right (anaglyph glasses required) delivers a more realistic representation of data. Minimally-invasive surgery, such as angioplasty or laparoscopy, can also benefit from stereoscopic vision. With the help from stereoscopic vision, doctors can achieve a 3D experience as if they were doing a full surgery (Salimpour et al., 1997; Salb et al., 2000; Ellis et al., 2005). In neuromuscular research, virtual reality environment created by 3D displays was shown to shorten balance recovery time of bedridden patients before they start walking again (Fig. 1·3.b). (Oddsson et al., 2006; Oddsson et al., 2007).

Stereoscopic vision can also be used in remote guidance (Olson et al., 2003). One such example is the recent mission of NASA to Mars (Bell III and *et al.*, 2003a; Ince and Konrad, 2005b). The rovers Spirit and Opportunity each carry a stereo camera which helps NASA scientists navigate the rovers precisely with the help of the depth information. Similar areas for stereoscopic vision are mining and operations in radioactively-contaminated areas that are potentially dangerous to humans (Konrad, 2001).

Simulators, such as for flight, driving etc. can be improved using stereoscopy (Ilgner et al., 2004). The trainee will be more aware of the situation with an enhanced display.

Businesses will benefit from stereoscopic displays, too. Companies can demonstrate their products in life-size using 3D displays. Similarly, virtual visits of homes, vacation sites can be made possible (Konrad, 2001).

One final application of stereoscopic vision is entertainment. Adding the third dimension to movies and computer games will add realism and be more pleasing to audiences. IMAX^{TM}, and more recently Real-D^{TM}, movie theaters are well-known for such an experience. Overcoming hardware and software problems, 3D TVs can replace ordinary TVs in the future (Lipton, 1994; Matusik and Pfister, 2004).

1.3 Is stereo really 3D?

Stereoscopic devices, as described earlier, can be used for many applications, however, there is a crucial question: are stereoscopic displays really 3D displays?

Unfortunately, although these displays can deliver the feeling of depth, this cannot be considered a real 3D experience. The missing part is the ability to modify the perspective on the screen when viewer changes his/her position.

In real life, when looking at an object, if we move our heads, we can see around the object. This is called the '*look-around feeling*'. Therefore, a realistic 3D device must be able to change the perspective if the viewer changes position. This is possible, for example, in volumetric displays because they create actual bright points in 3D space and the viewer is able to see different perspective by changing his/her position. However, in projected (planar) 3D devices, this is not directly possible, because in such devices, left and right views of the scene are painted on a flat screen and proper images are delivered to each eye by means of glasses or other devices. Therefore, when the viewer changes position, new images which will create the correct perspective must be painted on the screen. This inevitably brings the need for multiple images of the scene that are captured at different viewpoints as shown in Fig. 1·4.a.

Newly-emerging 3D displays solved the problem of changing perspective with respect to the position of the viewer. While some of these displays track the viewer either with a head-tracking device worn by the viewer or by cameras that capture the eyes of the viewer, others, such as displays with lenticular sheets, even removed this tracking requirement. These new displays brought us one step closer to a realistic and comfortable 3D experience.

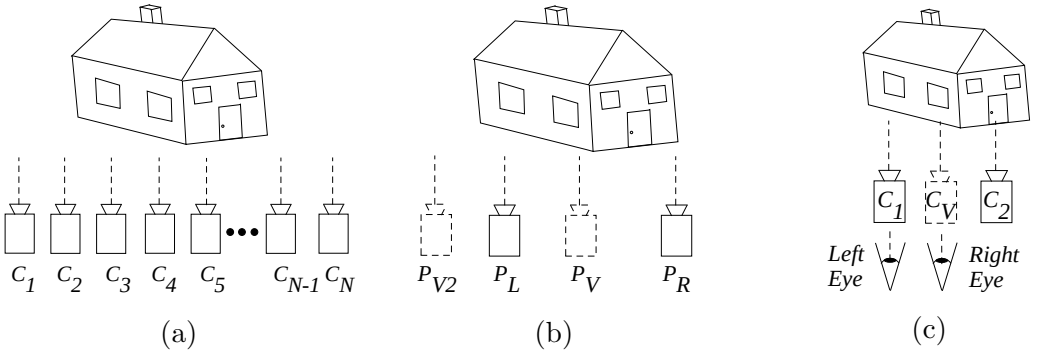


Figure 1·4: (a) Multiple images are required for a realistic 3D experience (b) Problem of intermediate view reconstruction. Given the images captured at positions P_L and P_R, is it possible to reconstruct the view at virtual position P_V? (c) Parallax on the screen can be adjusted by generating a new view for the virtual camera C_V.

However, the need for multiple images persists.

1.4 Elements of realistic and non-fatiguing 3D experience

Having distinguished stereo and 3D, we can conclude that there are two essential elements of a high-quality 3D experience.

Look-around feeling: When observing a static scene, if the observer changes position, the objects in the scene displace with different amounts; an effect called motion parallax. Therefore, the realism of a 3D system is only possible if the display can deliver motion parallax. Motion parallax, in turn, results in look-around feeling, which is the ability to see a changing perspective as the viewer changes position as mentioned in previous section.

In order to deliver motion parallax, a 3D system should have many images of the scene captured at multiple positions as shown in Fig. 1·4.a. Although displaying multiple images is possible by state-of-the-art technologies, generating multiple views of the same scene is still a problem.

Non-fatiguing viewing: Stereoscopic images/videos are captured by mimicking the human anatomy. Two identical cameras are positioned next to each other, similar to human eyes, and then the images from each camera (one for the left, one for the right

eye) are delivered to the viewer separately, e.g., glasses, lenticular sheets. However, the distance between the cameras is vital. If this distance is larger than the viewer's interocular distance, viewer will have difficulties with fusion of the left and right images. On the other hand, if it is smaller than viewer's interocular distance, the depth feeling will be subdued. Since it is not possible to shoot a movie for all possible interocular distances, the movies are shot for the average interocular distance, which in turn results in an uncomfortable experience for many people, for example children. The current technology, unfortunately, does not allow viewer to adjust '3Dness' of the scene.

1.5 Intermediate view reconstruction – remedy for 3D system issues

We mentioned two very important elements of a realistic 3D experience in previous section. The first one is the ability to show multiple views depending on the position of the viewer. This, of course, brings the need for many images of the scene as shown in Fig. 1·4.a. Using many cameras is the most trivial solution. However, this is not only an expensive and bulky solution but also requires extra effort, for example, to calibrate the cameras. Also, the number of cameras will always be limited. It is not possible to shoot a scene from every possible viewpoint. For example, if the viewer wanted to see the scene from a position between C_3 and C_4 in Fig. 1·4.a, the system would not have the corresponding data.

An alternative approach to this problem can be shooting the scene with few cameras (for example 2, 3 or 5) and then creating additional virtual views from the available images. Such techniques are called *intermediate view reconstruction* (or view synthesis or view interpolation or image-based rendering) and they constitute the main subject of this work. For example, the scene is captured by two cameras at positions P_L and P_R in Fig. 1·4.b. The question is what would the scene look like from the position of virtual cameras at P_V or P_{V2}?

By using intermediate view reconstruction, we can generate the views for cameras 2 to $N-1$ (Fig. 1·4.a) using only the images from cameras C_1 and C_N. This way, two cameras

will be enough to create the look-around feeling instead of a bulky rig with many cameras.

The ability to generate additional views with intermediate view reconstruction can also be used to enhance the comfort of a viewer when watching a 3D movie (Konrad, 1999). As mentioned in the previous section, a 3D experience with optimal comfort can only be possible if the acquiring cameras are positioned exactly at the interocular distance of the viewer. Intermediate view reconstruction can solve this problem by generating an additional virtual video stream or an image that is most pleasing for the viewer as shown in Fig. 1·4.c. The scene is shot using cameras C_1 and C_2, but a new view can be generated for the virtual camera C_V using the latter two.

As we elaborate on the background of an end-to-end 3D system in the next chapter, we will demonstrate additional applications of view reconstruction in 3D systems. In fact, view reconstruction has applications other than 3D systems as well, for example, in movie effects. Anyone who watched the movie *The Matrix* was amazed by the opening scene where the character Trinity jumps, time freezes and we see a full rotation of the camera around Trinity. This scene was shot by more than 100 cameras. Controlling and calibrating 100+ cameras is obviously a difficult job. Intermediate view reconstruction can generate this type of visual effects using a smaller number of cameras.

Intermediate view reconstruction can also be applied to a monocular sequence. For monocular sequences, intermediate view reconstruction can be used to change the frame rate by generating virtual frames between available ones. Intermediate views can also be generated in order to fill the missing frames in a video sequence. View reconstruction for monocular sequences will be discussed in Chapter 9.

View reconstruction is a challenging problem. Disparity estimation, handling of occlusion areas, formation of the intermediate view and intensity mismatches due to camera imperfections are some of the main problems. Later in this dissertation we will examine these challenges and offer novel solutions to each of them.

1.6 Outline of the dissertation

The outline of the dissertation is as follows:

Chapter 2 introduces the main concepts of stereo vision, disparity being the most of important of all. This chapter also describes components of a full end-to-end 3D system including human depth perception, acquisition, display and transmission of 3D data. Finally, this chapter outlines how view reconstruction can be utilized in every stage of a 3D system, therefore presents the motivation of this work.

Chapter 3 discusses prior work on intermediate view reconstruction. It starts by summarizing the assumptions and constraints of our work, then it categorizes the previous work with respect to the number of input images and effect of occlusions on each method and finally positions our work relative to other methods.

Chapter 4 is dedicated to one of the challenges; formation of intermediate view. It proposes overconstrained spline-based view reconstruction algorithm. Experimental results compare this method to the well-known pivoting-based method on ground truth data and prove its efficacy. It also discusses why this approach will be beneficial in handling occlusion areas.

Chapter 5 discusses the problem of occlusions from two perspectives. First, how to estimate occlusion areas and subsequently how to handle the occlusion areas once estimated. An occlusion estimation method is proposed to solve the first problem. Next, a disparity handling method, image-driven disparity inpainting, based on anisotropic diffusion is proposed to solve the second problem. Finally, this chapter elaborates on the interrelation of occlusions and disparity, and how these two problems can (and should) be solved jointly.

Chapter 6 is dedicated to disparity estimation. It starts by introducing optical flow and then proposes an occlusion-aware optical flow-based disparity estimation method. The proposed formulation jointly computes disparity, estimates occlusion areas and extrapolates disparity in occlusion areas. Experimental results show the superior performance of the method compared to state-of-the-art methods.

Chapter 7 combines all methods presented so far to create occlusion-aware spline-based view reconstruction. Experimental results demonstrate its efficacy especially in occlusion areas.

Chapter 8 focuses on extending the pivoting-based method to multiple images. The original pivoting-based method, which uses only two images, is unable to handle occlusions properly. Considering this problem, a new pivoting-based method that uses multiple images is proposed. Multiple images are essential to estimate and handle the occlusion areas. The proposed variational formulation adaptively uses different pairs of input images to estimate a disparity field and subsequently to reconstruct an intermediate view. Experimental results of proposed method show that the reconstruction quality in occlusion areas improves significantly.

Chapter 9 presents several applications of methods and ideas presented in this dissertation to real-world problems. First, proposed view reconstruction algorithm is used in health care as part of a special exercise bed for bedridden patients. Second, the view reconstruction algorithm is applied to monoscopic video sequences to improve the quality of videos captured by mobile phones. Third, a problem faced by NASA during Mars mission is solved by using proposed optical flow and view reconstruction algorithm. Finally, a block-based depth estimation algorithm is improved by using spatial regularization and used in a multiview video codec as a predictor.

Chapter 10 discusses contributions of this dissertation, draws conclusions and presents possible directions for future work.

Chapter 2

3D system design background

This chapter introduces system design issues of an end-to-end 3D system namely, perception, acquisition, transmission and display of 3D data. We will first discuss how humans perceive depth, and then we will present how stereoscopic images are acquired. Next, transmission of multiple image/video streams will be discussed. Then, we will classify the technologies used to display stereo. Finally, we will present applications of intermediate view reconstruction (IVR) in each of these steps, therefore presenting a part of our motivation for IVR.

2.1 Human perception: How do we perceive depth?

Depth is the relative distance of objects from the observer within a scene. Humans do not require any special effort to perceive depth of a scene. This is automatically achieved by the coordination of our eyes and brain. Our brain uses many cues to find the depth of an object. These cues can be classified as *monocular* and *binocular* depth cues.

2.1.1 Monocular depth cues

Monocular depth cues do not require the use of two eyes and as the name implies, they can be viewed with one eye only. We learn these cues starting from our childhood. Major monocular cues are as follows:

1. **Interposition (occlusion):** The most important monocular cue is the interposition of objects. If an object is blocking the view of another object, then it is obvious that the occluded object is further away from the viewer while the occluding object is

closer to the viewer. Figure 2·1.a shows a simple example; gray object is closer to the viewer.

2. **Geometric perspective**: Objects appear to be getting smaller as they get further away from the viewer. This effect is called geometric perspective (Franich, 1996). For example, the distance of points to the camera increases from A to B to C in Fig. 2·1.b.

3. **Light and shading:** Artists often use the lighting and shading in their works to emphasize depth. Shadows give an idea about the shape of objects and help us estimate the relative positions of points. Figure 2·1.c shows this effect (Lipton, 1991).

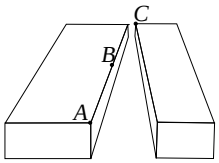
4. **Motion parallax**: This particular depth cue can be noticed during continuous movement of observer. In a static scene as the observer moves his/her head depending on the distance of objects from the observer, objects displace at different amounts. This effect can easily be noticed when traveling in a car. As the car moves, hills in the landscape move very slowly, while the traffic signs on the road pass by very quickly (Lipton, 1991).

5. **Relative size:** Another monocular depth cue is the relatives size of objects in the scene. Since our brain has a general idea about the sizes of objects in the world, we know that objects that appear larger are generally closer to us. Therefore, if two objects are known to have similar sizes and one of them is seen bigger than the other, then we know that the bigger one is closer to us. The house on the left in Fig. 2·1.d is closer to the viewer. This cue is very similar to 'geometric perspective' cue.

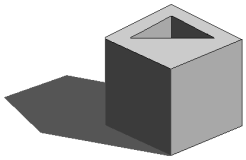
6. **Textual gradient:** Textual gradient depth cue is defined as the gradual change in the appearance of the object texture from coarse to fine (Lipton, 1991). The objects whose texture is more distinct appear closer. The object in the lower side of Fig. 2·1.e appear closer because of the textual gradient and geometric perspective.

7. **Aerial perspective**: The blurring of distant objects in the scene because of haze

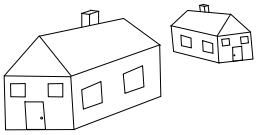
(a) Interposition



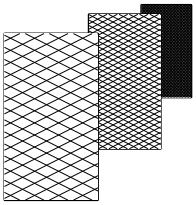
(b) Perspective



(c) Light and shade (Lipton, 1991)



(d) Relative size



(e) Texture gradient

(f) Aerial perspective

(g) M.C. Escher's "*Belvedere*"(1958)

Figure 2·1: (a)–(f) Examples for monocular depth cues (g) Famous artist Escher uses monocular cues to confuse the viewer. (*M.C. Escher's "Belvedere" ©2007 The M.C. Escher Company - the Netherlands. All rights reserved. Used by permission.* `www.mcescher.com`).

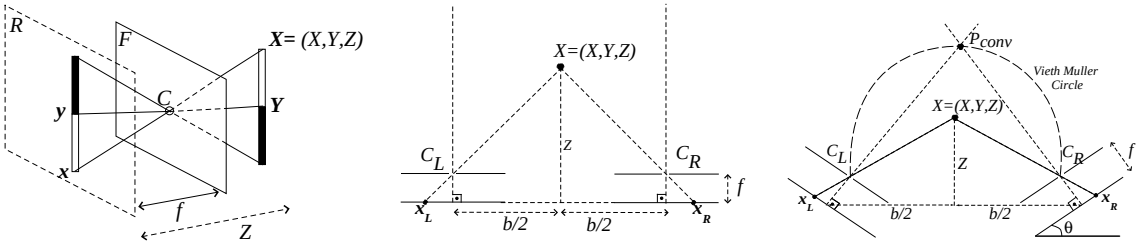
and scattering of light in the atmosphere is due to the aerial perspective. The objects in Fig. 2·1.f have the same sizes but the blurry one seems to be more distant.

The drawings of famous artist and mathematician Escher are best known to exploit the monocular depth cues to confuse the viewer (Franich, 1996). One such example (Belvedere, 1958) is shown in Fig. 2·1.g.

2.1.2 Binocular depth cues

Binocular depth cues, unlike monocular ones, are perceived by two eyes. Since human eyes are positioned about 2.5 inches apart, projections of a scene onto retinas of eyes are slightly different. The brain fuses these two different images to perceive the depth of the scene. This is often referred to as *binocular stereopsis*.

In order to understand this effect, one can do this easy experiment: While looking at a



(a) Pinhole camera model (b) Parallel cameras (c) Toed-in cameras

Figure 2·2: Two commonly used stereo acquiring configurations.

stationary scene, close and open one eye at a time. It is easy to notice the differences when looking with left or right eye. The main difference between both eyes' perceptions is that positions of objects are different. This displacement between projections of the same 3D point in left and right images is called *disparity*, a very important concept in stereo vision. The brain uses disparity information in order to find depth.

The disparity is larger for objects that are closer to the viewer, in other words, disparity is inversely proportional to depth. One can do the previous experiment to observe this fact, too. Put your thumb about 5 inches away from your eyes. While looking at a distant object, close and open one eye at a time. It is easy to notice that your thumb 'moves' quite a bit, while the distant object 'moves' very little. As mentioned before, disparity is a very important concept in stereo vision because it gives the depth information of an image point. This will be elaborated upon when we discuss camera structures next.

2.2 Acquisition: Camera models

2.2.1 Pinhole camera model

Before discussing multiple-camera setups, we would like to first introduce *pinhole camera model*, the simplest camera model. Although very simple, this model describes the geometry and optics of most modern cameras quite accurately (Faugeras, 1993).

Acquisition of an image using a pinhole camera model is illustrated in Fig. 2·2.a. Light rays departing the scene pass through a small hole, C, called the *center of projection* or *optical center*, which resides on the focal plane, F, and fall onto the projection plane R to

form an inverted image. The distance between planes F and R is called the *focal length* and denoted by f. The focal length controls the size of the projection of the object. This is why a 100mm lens has more zooming properties than a 35mm lens. Equations describing projection of a point onto image plane are derived in Appendix A. We would like to refer the reader to a book by Faugeras (Faugeras, 1993) for an extensive analysis of camera models.

As we mentioned, stereo images are captured by two identical cameras mimicking the human anatomy. There are two primary camera configurations used in stereoscopic vision. The *parallel camera configuration* is composed of two cameras with *parallel* optical axes (Fig. 2·2.b). This configuration is often used because of the simplicity of its mathematical derivations. The second configuration is *toed-in* or *converging camera configuration* (Fig. 2·2.c). In this configuration, cameras are rotated towards each other by a small angle, so that the optical axes of the cameras converge at some point other than infinity.

2.2.2 Parallel cameras

Parallel cameras have parallel optical axes which intersect at infinity. The horizontal distance between the optical centers of the two cameras is defined as the *baseline distance* (denoted by b in Fig. 2·2.b). Consider the point $\mathbf{X}$ in 3D world with coordinates (X, Y, Z) and its projections $\mathbf{x}_L$ and $\mathbf{x}_R$ on image sensors of the left and right cameras. Assuming that two identical cameras with identical zoom settings are used, let the focal length of the cameras be f. Disparity $\mathbf{d}$ is then defined as

$$\mathbf{d} = \mathbf{x}_L - \mathbf{x}_R = \begin{bmatrix} -f\frac{b}{f-Z} \\ 0 \end{bmatrix}. \tag{2.1}$$

A detailed derivation for $\mathbf{x}_L, \mathbf{x}_R$ and $\mathbf{d}$ can be found in Appendix A. The disparity vector indicates that there is no vertical shift between the images, the most important property of parallel cameras. The Z component, which is the depth of the point $\mathbf{X}$, can be computed

using similarity of triangles $(\mathbf{X}, \mathbf{x}_L, \mathbf{x}_R)$ and $(\mathbf{X}, C_L, C_R)$ in Fig. 2·2.a as follows:

$$Z = f\frac{b+d}{d}. \tag{2.2}$$

The advantage of this setup is that the geometric relations governing parallel cameras are simple. Moreover, the vertical component of disparity is always zero. This property is very valuable for computational simplicity of disparity estimation; homologous points must lie on the same image scan line. On the other hand, one of the disadvantages of this setup is that since the optical axes of the cameras are parallel to each other, they will converge at infinity, therefore there are not going to be any points in the images whose disparity is zero. Due to this, stereo images captured by the parallel camera setup can demonstrate excessive disparities, especially for closer objects. Viewing of such stereo images will not be comfortable. Another problem is that for larger baselines the common field of view is small. This creates problems for disparity estimation methods as these methods match points in one image with points in the other image.

2.2.3 Toed-in cameras

The projection of a 3D point in a toed-in camera setup is more complicated than in the parallel cameras case. A full derivation can be found in Appendix A. There is an interesting geometrical concept for toed-in cameras. A 3D point that lies on the circle called Vieth-Müller circle (Franich, 1996), (Fig. 2·2.c) which passes through the optical centers of cameras and the convergence point P_{conv}, has zero disparity after projection onto the image planes.

The advantage of toed-in setup is that it creates positive and negative parallax in captured images. Therefore the absolute value of maximum or minimum disparity is usually smaller than that of parallel cameras, which in turn leads to more comfortable viewing experience. This can also be used to create desired 3D effects e.g., object can be perceived as 'sticking' out of the 3D display. Moreover since the optical axes of cameras intersect at a physical point, the common field of view is increased compared to parallel cameras. On

the other hand, the analysis of geometry is much more complicated than that of parallel cameras. Another disadvantage is that this configuration introduces opposing keystone distortions in the stereo pair. Keystone distortion can be described as a distortion that projects a rectangular frame onto a trapezoid. Finally, disparity has a vertical component and therefore must be represented by a two-dimensional vector. This, in turn, brings extra computational complexity into the disparity estimation methods.

A detailed comparison of toed-in and parallel camera configurations along with resulting distortions can be found in the paper of Woods *et al.* (Woods et al., 1993). Subjective evaluations of each setup can be found in the paper of IJsselsteijn *et al.* (IJsselsteijn et al., 2000)

2.2.4 Camera arrays

Recent computer graphics and vision applications extended two camera setups to many cameras, usually called camera arrays (Moravec, 1980; Wilburn et al., 2002; Matusik and Pfister, 2004; Wilburn et al., 2005). The number of cameras can range from 8 up to 128 cameras. Impressive applications are created using these setups, however special hardware is required to control and synchronize the cameras. The geometrical relationship between pairs of cameras in an array can be derived by extending parallel camera setup after including location of cameras in 3D space.

2.3 Communication: Data transfer for multiview displays

Transmission of multiview data is another major obstacle in a 3D system. Although efficient compression methods are available for a single video sequence, transmission of multiview data is a new research area. The trade off here is that increasing the number of cameras improves the quality of 3D experience at the expense of increased transmission bandwidth.

The trivial solution would be to compress multiple video streams independently of each other, however, since the redundancy between video sequences is not exploited, this would not be an efficient technique (Smolic and Kimata, 2003; Vetro et al., 2004). Considering

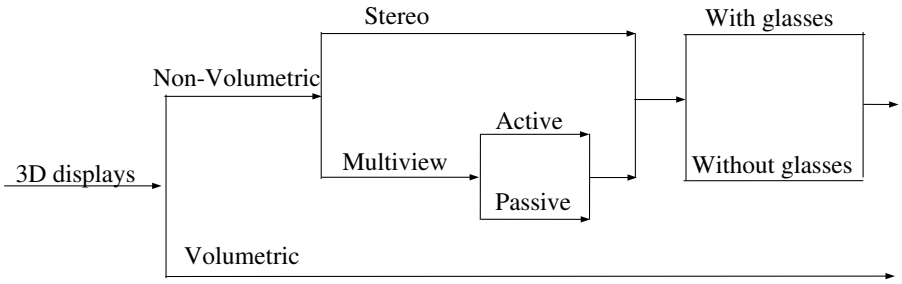


Figure 2·3: General classification of stereo systems.

this problem, there is currently an MPEG activity to standardize compression of multiview data. Possible solutions include reconstructing intermediate views (Kimata and Kitahara, 2004; Martinian et al., 2006a; Ince et al., 2007b) to increase compression efficiency (to be discussed in Section 2.5) or exploiting advanced prediction orders in existing video coders (Merkle et al., 2006).

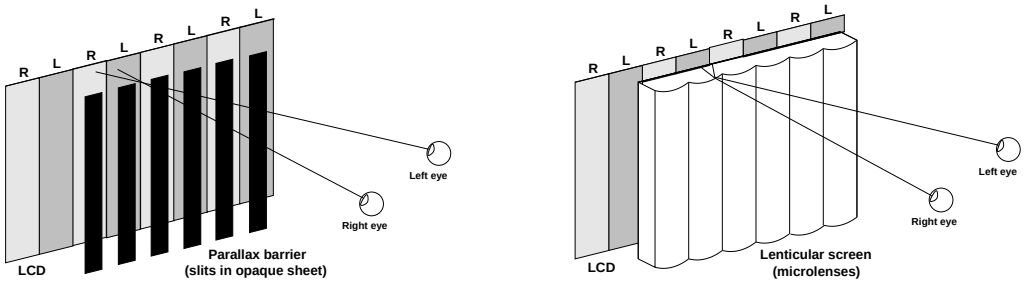
2.4 Display: Classification of 3D displays

Having discussed the perception, acquisition, and, finally, transmission of stereo images, let us now explain how these images are presented to the viewer so that he/she can perceive depth. There are many different types of 3D displays. We will present only a coarse classification. More detailed and different types of classifications can be found in (Benton, 2001; Konrad and Halle, 2007)

As shown in Fig. 2·3, 3D displays can be divided into volumetric and non-volumetric (projected or perceived) displays. Volumetric displays create actual bright points in a 3D volume explicitly, thus mimicking a 3D structure. Two examples of volumetric displays are rotating screens and multiple semi transparent screens (Sullivan, 2005).

On the other hand, non-volumetric displays create the 3D perception by multiplexing many images projected onto the same screen. Since the results of our work apply to non-volumetric displays, we will concentrate on such displays from now on.

A non-volumetric 3D display can be either stereo (2 views) or multiview. The multiview displays deliver a more complete 3D experience since they are able to show different



(a) Parallax barrier (b) Lenticular sheet

Figure 2·4: Autostereoscopic displays.

perspectives of the scene as the viewer changes his position. In real world, when we are looking at a static object, if we move our heads we can see around objects due to motion parallax cue discussed earlier. This so-called *'look around feeling'* is possible on a 3D display only if it can deliver multiple views. On the other hand, stereoscopic displays show the scene from only a single viewpoint. If the viewer changes his position, he/she always sees the same images, leading to the impression that the scene rotates, a clearly unrealistic experience.

Since both left and right images of the scene are shown simultaneously on the screen, a separation step is needed to deliver proper image to each eye. Many 3D systems solve this problem using glasses that the viewer must wear.

There are many examples of such glasses. Anaglyph glasses have a red lens on the left eye and a blue (sometimes cyan) lens on the right eye. Anaglyph stereo images are produced in such a way that two images, a red and blue image, are superimposed through red and blue channels of a color image. Since anaglyph glasses have colored lenses, the left and right eyes see the red and blue images respectively. Since two different images are perceived by each eye, viewer can perceive depth. Anaglyphs offer acceptable quality for black and white images, however, a true color image is not possible since viewing is through the colored glasses (McAllister, 1993).

Another example are polarization-multiplexed displays. In this setup, two views are projected through light polarizers onto the screen by using two projectors. These superim-

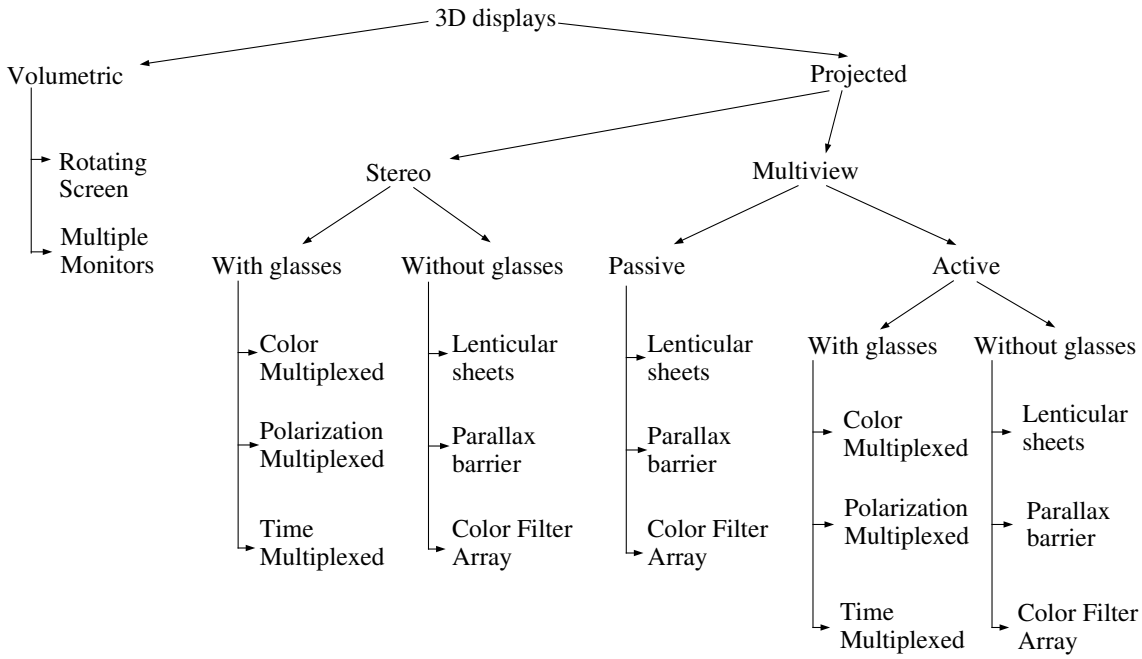


Figure 2·5: Classification of 3D displays with some examples.

posed images are then separated by polarized (circular or linear) glasses worn by the user (Pastoor and Wöpking, 1997). Alignment of the projectors is a challenge for this kind of displays.

One final example is the time-multiplexed displays which exploit the fact that human visual system can retain an image for some time. The left and right views are sequentially shown on the screen and lenses of liquid crystal shutter glasses open and close in synchronization with the displayed images. When the left view is displayed, the right lens occludes the right eye and vice versa. The operation of glasses is controlled by an infrared emitter placed close to the monitor or the glasses are directly connected to the display (Pastoor and Wöpking, 1997).

Recently, researchers eliminated the necessity of eyewear. Examples of systems without eyewear include parallax barrier (Fig. 2·4.a) and lenticular sheet (Fig. 2·4.b). Parallax barrier is a thin opaque material with a series of regularly-spaced vertical slits (Halle, 1997). Each slit acts as a window to the stripe of the image behind the parallax barrier.

The visibility of a strip depends on the horizontal viewing angle. A stereoscopic image is displayed by interleaving columns of the two images. If the viewer is positioned at the appropriate location, the left and right eyes will see different images, therefore depth will be perceived.

The other type of monitors uses lenticular sheets (sometimes called lens sheets or micro lenses) which can be regarded as many miniature lenses arrayed on a flat sheet. This sheet covers the surface of a flat panel monitor. The screen and the lenticular sheet are precisely aligned so that at a specific position of the viewer, the left and right views are properly delivered to the viewer's eyes as a result of diffraction of light.

As we said, a multiview display can deliver different perspective if the viewer changes position. Some systems use a tracking device to track the position of the viewer and they are called active systems. This type of systems includes wearing a head tracker or using software that tracks the eyes of the viewer. As the position of the viewer changes, the system adjusts to the new position by either painting new images on the screen or sometimes even by changing the physical structure of the display. On the other hand, passive systems do not track the position of the viewer. They accommodate the change in the position of the viewer by displaying different views at different positions *simultaneously*. Obviously, active systems cannot be used by two viewers at the same time since the system cannot adjust two different positions at the same time. Passive systems, such as lenticular sheets, let many viewers use the same screen effortlessly, but with reduced spatial or temporal resolution due to the view multiplexing. However, the availability of very high resolution displays remedy this problem.

After giving information about several different types of 3D systems, we show a classification in Fig. 2·5.

2.5 Applications of IVR in an end-to-end 3D system

Having discussed the individual components of a 3D system, let us now summarize the applications of IVR in a 3D system. As mentioned in Chapter 1, IVR can be used in all

steps of an end-to-end 3D system.

Acquisition and perception stages: As mentioned in the previous chapter, the comfort of the viewer is highly dependent on the amount of parallax, therefore, the distance between cameras. Intermediate view reconstruction can be used to virtually adjust the distance between cameras so that a viewer finds the most comfortable position, as shown in Fig. 1·4.c. It is possible to consider this as a '3Dness' knob much like the volume knob found on TV sets (Konrad, 2001).

Communication stage: The goal of video compression is to eliminate redundancy, by predicting the current data from previously-transmitted data. Therefore, high-quality references are needed to effectively eliminate the redundancy. Intermediate view reconstruction can be used to generate additional reference pictures by using available cameras. For example, in Fig. 1·4.a, cameras C_1 and C_3 can be used to create an additional reference for C_2 to increase compression efficiency.

Similar to display stage, it is also possible to reduce the number of cameras by using view reconstruction. It may be possible not to transmit any data regarding C_2 in Fig. 1·4.a and save significant amount of bandwidth. C_2, then, can be reconstructed in the decoder side by using C_1 and C_3.

Display stage: As pointed out in the classification of 3D displays, current technology eliminated the need for glasses. However, these displays require many images of the scene (usually around 10 individual images). One approach to solve this problem is to use camera arrays, however this may be an expensive solution. Intermediate view reconstruction can generate multiple views using a small number (2-3) of cameras.

Similarly, it is possible to migrate vast amount of historical stereo images to new multiview 3D displays by generating additional views for these images.

Finally, intermediate view reconstruction can offer a scalable 3D broadcast. If the number of cameras in the acquisition step is not equal to the number of required input images of 3D display on the receiver side, then additional views can be generated on the receiver side so that requirements of the display are satisfied. This, of course, brings a

computational load to the decoder but as the processing power increases, this would be less of a concern.

2.6 Conclusions

In this chapter, we presented basics of an end-to-end 3D system, an example of which can be found in (Matusik and Pfister, 2004). After describing required steps and challenges in such a system, we outlined many applications of view reconstruction to solve these problems. It is clear that view reconstruction can improve 3D systems and enhance the viewer experience.

Chapter 3

Intermediate view reconstruction: state-of-the-art and challenges

This chapter reviews the prior work on intermediate view reconstruction. However, it first starts by stating assumptions and envisioned applications of our work so that it will help us position our work with respect to prior art.

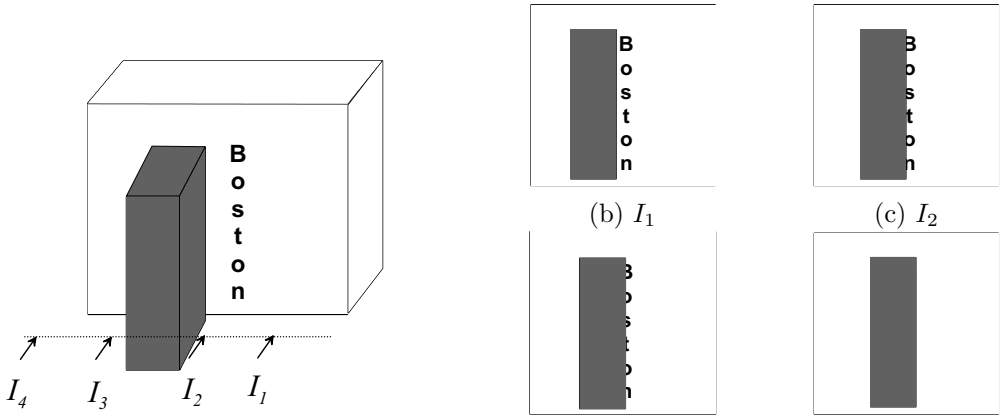
Computer graphics, computer vision and image processing communities have proposed many solutions to IVR to date. We will give a classification of past methods depending on their requirements vis-a-vis the number of input images and their need to know camera geometry. We will point out the occlusion-awareness of algorithms as well. The challenges in view reconstruction, which are going to be elaborated upon in more detail in the subsequent chapters, will be briefly introduced.

3.1 Applications and constraints of this work

Considering the emergence of automultiscopic (i.e., multiview, eyewear-free) displays, our work envisions the following applications:

- Migrating available stereo images to multiview displays. Similarly extending readily available stereo acquisition setups to multiview displays.
- Content generation for multiview displays using small number of cameras.
- Applications where using a large number of cameras is unrealistic such as medical applications or space exploration.

The constraints and assumptions in this work will be as follows:



(a) (b) I_1 (c) I_2 (d) I_3 (e) I_4

Figure 3·1: A schematic illustration of occlusions. (a) Images of a 3D scene are captured at four different locations generating four images I_1 to I_4 shown in (b)-(e). The gray object gradually occludes the text from I_1 (first image) toward I_4 (last image). The intermediate images I_2 and I_3 show the partial occlusion of text.

- We will mainly focus on using two input images. There must be at least two input images for a stereo application. If the proposed approaches can successfully handle two input images, extension of algorithms to multiple cameras will be trivial by choosing any pair of images. However, for the sake of completeness, we will focus on multiple images in Chapter 8 as well.

- We will assume that the images are captured by small/medium-baseline cameras which corresponds to 6-7 times of interocular distance. Larger baseline distances are not suitable for multiview displays since humans cannot fuse such images.

- We will be flexible on camera geometry restrictions. Most methods can work solely on parallel camera structure which is hard to achieve. Methods proposed in this work will allow flexible camera geometry. However, the generated views will be on a line connecting the focal points of the real cameras, again due to the specific application of multiview displays.

- We will focus on images captured by uncalibrated cameras. The reason is that stereo images usually do not have the required calibration data. Related to this, we will

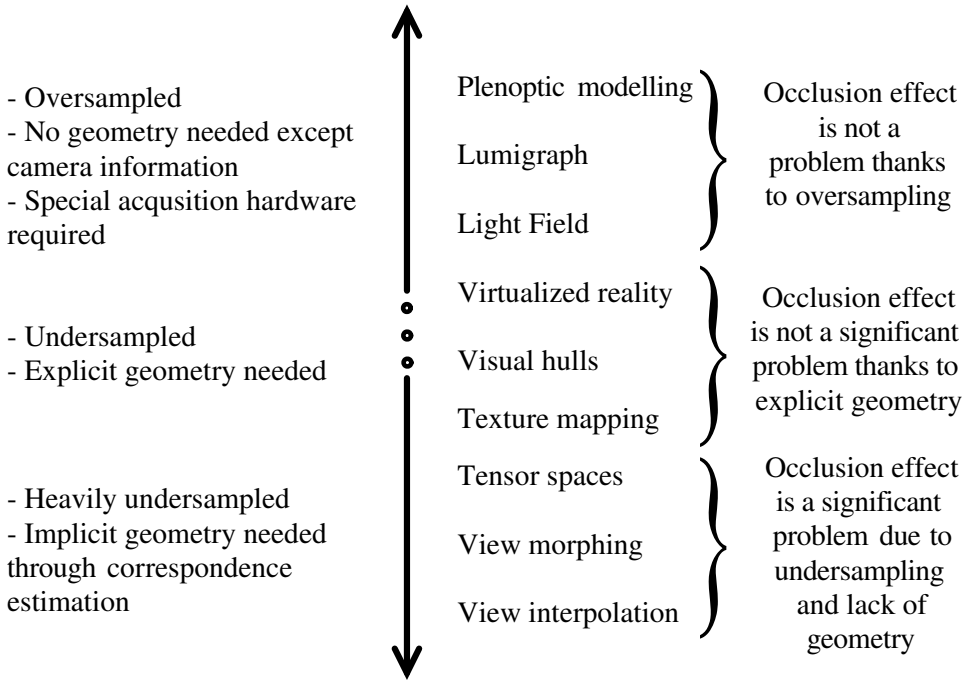


Figure 3·2: A rough classification of view reconstruction algorithms based on the requirements of number of images required, geometric information and effect of occlusions. Required number of images decreases from top to bottom.

not use rectification which practically introduces blur in the reconstructed images.

The main focus and effort in our work is to work with small number of images and to focus on the occlusion areas. The occlusion effect is illustrated in Fig. 3·1. Assume that images of a 3D scene, shown in Fig. 3·1.a, are captured at four different locations denoted by arrows in the image. Since the gray object is closer to the camera, it will have a larger disparity value than the text in the background. Therefore, as the gray object in I_1 displaces between images and it will gradually occlude the text. In I_4 the texture is completely occluded while intermediate views demonstrate partial occlusion. The reconstruction in occlusion areas is particularly difficult as will be explained in subsequent chapter. Now let us review prior work in the area.

3.2 Prior work on view reconstruction

Intermediate view reconstruction (or image based rendering) is basically creating an image at a specific position at a specific time. Adelson and Bergen (Adelson and Bergen, 1991) formulated all possible images by the so-called *plenoptic function* that records all light rays at every possible 3D location (V_x, V_y, V_z) that is looking towards every possible direction (θ, ϕ) at every time t for every wavelength λ. Considering this seven-dimensional function, $P(V_x, V_y, V_z, \theta, \phi, \lambda, t)$, generating new images means simply sampling this function. However, due to its many dimensions and complexity, capturing a full plenoptic function is difficult, if not impossible. Researchers usually make assumptions to reduce the dimension of function P. For example, if we consider a static scene (i.e., fixed time) and assume grayscale images (i.e., fixed wavelength), the number of dimensions immediately reduces to five. It is possible to classify prior work depending on the number of dimensions of the plenoptic functions used as in the paper by Zhang and Chen (Zhang and Chen, 2004). However we would like to classify the methods depending on their need for geometry information (similar to (Shum et al., 2003)) and, more importantly, the number of required input images. This will also help us analyze the methods from the occlusion-awareness point of view.

A coarse classification is shown in Fig. 3·2. The number of input images required by the methods decreases from top to bottom.

Category #1: Methods that rely on oversampling: The first type of methods such as *lightfield rendering* (Levoy and Hanrahan, 1996) and *lumigraph* (Gortler et al., 1996) rely on oversampled data of the scene. In both works researchers create a 4D representation of the scene using many input images. The intermediate views are then created simply by slicing (sampling) this 4D representation, i.e., 4D plenoptic function. The input images are captured by regularly spaced cameras on a 2D array. The difference between lumigraph and light field rendering is that lumigraph can use images taken from arbitrarily placed cameras by using special markers in the scene. However, these images are eventually

projected onto a regular array of images, a process called re-binning by the authors.

There are no assumptions made about scene geometry, however if the number of images is small, then artifacts are observed in the reconstructed images. Another problem is that the illumination must be fixed and cannot change during acquisition. However, the main disadvantage of this method is its need for many input images as many as thousands.

Since the scene is oversampled, the rendering process in both of these approaches is independent of the geometry and simply blends input images. The presence occlusions is not a problem because, thanks to oversampling, occlusions between nearest cameras are negligible and all texture in the scene is visible in at least a few cameras.

A multiple camera system was also proposed by Kanade *et al.* (Kanade et al., 1997) to create intermediate views for dynamic scenes. The input images are acquired in a specially built dome which consists of 51 cameras. The main limitation of the method is its need for a special acquisition step; a multi camera dome.

Category #2 : Methods that use undersampled data sets with available geometric information: Given the geometry of a scene, it is possible to reduce the number of images needed. Such a method was proposed by McMillan (McMillan, 1997). If the depth (equivalent to disparity) of the input images is readily available, it is possible to project pixels of the original images to a new viewpoint and reconstruct a new image. Obviously, it is not guaranteed that all pixels in the new image will be visible in the input images. Therefore, occlusions should be handled in some way. However, since the geometry of the scene is known, at least locations of occlusions are known.

In another approach, called view-dependent texture mapping, given scene geometry and scene texture, it is possible to render new images. Debevec *et al.* (Debevec et al., 1998) proposed such a method that first creates a 3D model of the scene and then maps the texture onto this model. Camera geometry and camera locations are assumed to be known. The results of the method are shown for aerial pictures where objects have well-defined geometric shapes (e.g., buildings), however arbitrary objects may pose difficulties. For example, only the buildings are modeled while smaller objects such as trees are not

considered in the 3D model.

Similarly, Buehler *et al.* (Buehler et al., 2001) proposed a method, called *unstructured lumigraph rendering*, which is a generalization of some image based rendering algorithms. The method becomes a lumigraph-style approach in one extremity (lack of geometry with many input images) while it behaves like view-dependent texture mapping (small number of input images with scene geometry). The advantage of the method is its ability to generate a virtual view when presented with different number of input images and camera geometry.

Another method, which uses small number of cameras, was proposed by Matusik *et al.* (Matusik et al., 2001) to recover a 3D shape of an object by computing a visual hull. A visual hull is the intersection of projections of object boundaries. Due to the nature of projections, visual hull cannot contain any concavities, which is a major limitation. Also, object boundaries are extracted using segmentation methods which are prone to errors. These errors are visible around object boundaries.

As we said, in these methods occlusions are a problem to a degree because they must be handled carefully. However, thanks to the known geometric information, at least occlusion areas can be detected in advance.

Category #3: Methods that use heavily undersampled data sets with unknown geometry: In the final category that we are considering are the methods that have no access to the scene geometry and work on a small number (2-3) of input images. Therefore, they use heavily-undersampled data sets. Most of these methods do not use any camera calibration information either. These methods implicitly compute the geometry (usually from disparity) of the scene either using correspondence matching or projective geometry. The work presented in this dissertation is closest to this type of methods.

These methods can be categorized based on two criteria. The first criterion is whether these methods compute disparity by using backward- or forward-projection. In backward-projection, disparity and intensity of each pixel of the intermediate view are estimated by pivoting on the *unknown* intermediate image and by back-projecting this pixel onto the known images to extract disparity and intensity information. In the complementary,

forward-projection approach, the disparity is estimated on the *known* input images and then intensities of input images are projected onto the intermediate view to reconstruct the intermediate view. These two approaches will be discussed in detail in Chapter 4.

Here, we would like to classify the methods based on a second criterion: the type of disparity method used to infer the geometry of the scene.

Projective geometry-based methods: Two examples that rely on projective geometry are works of Seitz and Dyer (Seitz and Dyer, 1996) and Avidan and Shashua (Avidan and Shashua, 1997). Seitz and Dyer proposed a view morphing algorithm to generate a morphing between two available images which can also be used to generate intermediate views. The method is composed of three main steps: First, original images are rectified and then two intermediate views are reconstructed using projective geometry, which is computed using feature points. The final intermediate view is generated by blending these two images using weighted averaging. The third and final step is to inverse-rectify the reconstructed image into the desired viewpoint. Although being an effective method, it suffers from the low-pass filtering effect of the rectification steps. Moreover, since occlusion areas are not handled explicitly, these areas demonstrate a 'ghosting' effect during transition. Exposed areas in the intermediate view are filled using texture synthesis; an implicit property of the method. However, efficacy of texture synthesis is limited.

Avidan and Shashua (Avidan and Shashua, 1997) utilized *tensor spaces* to create intermediate views. Point correspondences between input images are used to compute a trilinear tensor, which can be considered as a mapping from reference images to the new image. Then, this tensor and optical flow information between available images are used to generate virtual views. Occlusions are noted as one of the major problems by authors as the method is unaware of the visibility of points.

Scharstein (Scharstein, 1996) combined rectification with disparity estimation to handle occlusions. The method computes a disparity field between input images and then creates two intermediate views at the same position by forward mapping (i.e. disparity compensating the images using disparity vectors) both the left and the right images. The

pixel positions are rounded to nearest integer, a process which degrades quality of images. Newly-exposed areas are filled by using either texture of one of the frames or texture synthesis algorithms. Occlusions, which the author calls overdefined points i.e., more than two points in the original image falling on the same location in the intermediate view, are handled by ordering the depth of points. This method being simple, suffers from the forward mapping part. Filling the holes using texture synthesis is problematic for reconstructed views, especially in areas with detail.

Optical flow (disparity) based methods: Actually, creating virtual views using the optical flow between input images dates back to the work of Chen and Williams (Chen and Williams, 1993) where they proposed a method to generate multiple view of a scene using a few closely spaced viewpoints. The idea was to compute correspondences between images and create a viewpoint using these correspondences. Their method focused on synthetic images. Occlusions are not effectively handled as they use texture synthesis algorithm to generate parts of the frame. This is the most similar method to the method that we envision. However, we are considering real images that requires the extraction of both depth and occlusion information.

Most recently, Zitnick *et al.* (Zitnick et al., 2004) proposed a full system for view synthesis of dynamic scenes. Their system is composed of 8 calibrated cameras placed on a line. The system first segments all images using color information and then computes the depth of the scene using 3 neighboring cameras. Original videos along with disparity maps, boundaries of objects and matting information are all stored using a specially-built multi-view encoder. The view generation is achieved in real-time.

Block-based methods: Block-based techniques have been used in the context of intermediate view reconstruction. However, their inherent assumption that all points in a block should have the same disparity does not always hold. Therefore, it is suggested to change the block size when required. Mancini and Konrad (Mancini and Konrad, 1998) proposed a quadtree block matching technique which first calculates the disparity values for larger blocks and then reduces the block size at possible boundary locations. The reconstruction

of intermediate view is achieved by pivoting-based method which is discussed in Chapter 4. However, occlusions are not addressed by Mancini and Konrad.

In another block-based approach, McVeigh *et al.* (McVeigh et al., 1996) explicitly detect occlusion areas and handle them by assuming that depth stays constant within the neighborhood of occlusions. The equations used for formation of the intermediate view indicate that they use full-pixel precision to avoid irregularly-spaced intensities, which is the main limitation of the method.

Pivoting-based (back-projection-based) techniques use linear interpolation (weighted averaging of intensities at endpoints of the disparity vector) methods which tend to result in blurry intermediate views. Mansouri and Konrad (Mansouri and Konrad, 2000) proposed a winner-take-all approach to overcome this problem. In their approach, the intermediate view is reconstructed by tilings from either left or right images. In other words, every block in the intermediate view is equal to the corresponding disparity-compensated block in either the left or the right image. Although this method decreases blur in the reconstructed image, it also introduces a "patchiness" effect when left and right views have significant differences in intensities. Although using either left or right image hints a method of handling occlusion areas (because occlusion areas are visible in either image), authors do not address occlusions explicitly.

Dynamic programming-based methods: Redert *et al.* (Redert et al., 1997) introduce a method which can reconstruct views at *non* intermediate positions. Their motivation is to overcome the restriction that fixes the new viewpoint between the cameras. They compute disparity fields using dynamic programming. Then, they compute the intermediate view at the center point between the two cameras. For the rest of the algorithm, they use this center view and a single disparity map D (either right-to-center or center-to-left). Image rendering using this method also suffers from problems like in Scharstein's case. Occlusions are handled via depth ordering of points, while newly-exposed areas are filled in by linear interpolation of available intensities, which can offer only limited quality.

Feature-point-based method: Siu and Lau (Siu and Lau, 2005) propose an image regis-

tration technique for view rendering. Their aim is to reduce the number of required images for the matching step. Their method, similar to (Kardouchi and Konrad, 2003), first extracts feature points using the Harris operator and matches these points between images. Unlike our method, they use three images to verify correspondences. This step is followed by Delaunay triangulation and topological consistency checks. Once a full disparity field is computed, a virtual view is reconstructed. A deficiency of this method, which can be observed in their results, is the 'ghosting effect' (blurring of the edges) in the reconstructed images which is especially significant in textured areas that are being occluded/exposed. This is an indicator of unsuccessful handling of occlusion areas. Also, due to Delanuay triangulation, areas closer to boundaries of the images cannot be reconstructed properly. Resulting images are cropped such that the boundaries of original images are excluded.

Multi-camera methods: Utilizing more than two cameras for view synthesis was also proposed in the literature. The advantage is that occluded parts may be better defined using additional views. Park and Inoue (Park and Inoue, 1997) proposed an arbitrary view generation algorithm using five cameras. Their camera system consists of a center camera and four additional cameras (above, below, left and right) separated by the same distance. In their algorithm, they exploit a depth map of the central camera computed assuming texture in this camera is visible in other cameras. They forward map the image of the central camera to that of a virtual camera. However, the problem of having overdefined or undefined points again arises due to the forward mapping and due to occlusions. They use various assumptions such as depth constancy to fill in the occlusion areas. In the event of the failure of all assumptions, they resort to texture synthesis that blends colors of neighboring positions, which is another deficiency of their method because texture synthesis algorithms often fail in high-detail areas.

As we mentioned in the beginning of this category, all these methods reviewed in category use either forward (e.g., Redert *et al.*, McVeigh *et al.*, and Scharstein) or backward (e.g., Mancini and Konrad, Mansouri and Konrad) projection to reconstruct images. It should be noted forward projection based methods such as Redert *et al.*, McVeigh *et al.*, and

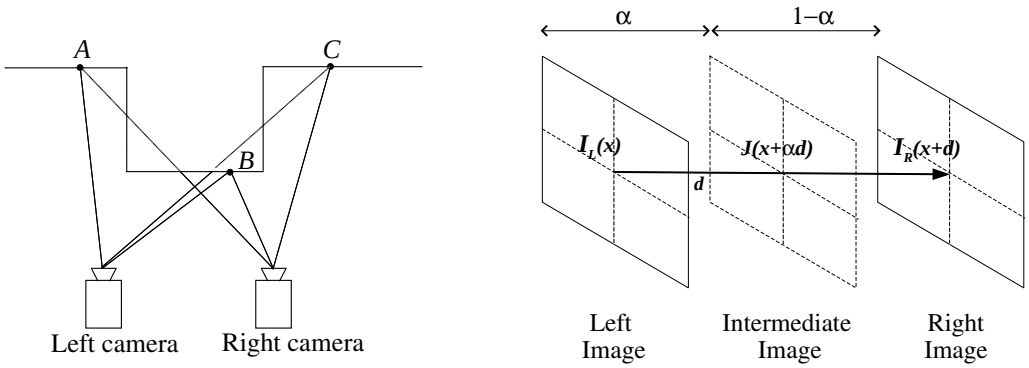
Scharstein share common features with the work of McMillan (McMillan, 1997) where depth was readily available. More recent work eliminated the challenge of disparity estimation by using special cameras that record not only the photometry but also depth information. This so-called 'depth-image-based rendering' became a reference tool for new 3D displays from Philips (Redert et al., 2002).

When compared to the previous two categories we presented, the occlusion effects are most problematic in this category for two reasons. First of all, geometry (depth, disparity or projective mapping) should be extracted from the images, and secondly these methods are uninformed about occlusion areas. Therefore, they should properly detect and handle the occlusion areas. Moreover, as we will discuss later in this dissertation, since occlusion areas cause significant problems in correspondence estimation stage, these algorithm should pay close attention to the such areas. As we mentioned, our work falls into this category as well.

Overall, the main problems in prior algorithms are as follows:

- Occlusions are not handled or they are handled in a simplictic way.
- Full-pixel precision is used to avoid irregularly-spaced intensities.
- Forward-projection based methods use simple approaches (usually nearest neighborhood interpolation) to convert irregularly-spaced data to regularly-spaced data.
- Many input images are required.
- Additional input parameters such as focal length, baseline distance etc. are required.

Finally, there is an important point that we would like to emphasize. The results of view reconstruction are usually evaluated subjectively. Since the underlying true image is not always available, the quality of reconstruction cannot be assessed numerically, e.g., with mean square error. In this work, by using data sets that contain many views of a scene, we will try to present reconstruction results with Peak-Signal-to-Noise-Ratio (PSNR) values.



(a) (b)

Figure 3·3: (a) Illustration of occlusion areas. Point B is visible in both cameras while A is visible only in the left-camera image and C is visible only in the right-camera image (b) Disparity estimation method creates irregularly-sampled intensities in intermediate view.

PSNR, a measure of mean square error, of two images is defined as follows:

$$\varepsilon_{MSE} = \sum_{x=1}^{N_x}\sum_{y=1}^{N_y} \frac{\left(I(x,y) - \widehat{I}(x,y)\right)^2}{N_x \times N_y}, \quad PSNR = 10\log(\frac{255^2}{\varepsilon_{MSE}}), \tag{3.1}$$

where N_x and N_y are horizontal and vertical dimensions of the images. As it can be seen from this equation, higher PSNR values indicate smaller reconstruction error and therefore, higher quality images.

3.3 Challenges in view reconstruction

Reviewing prior work in the field, we can conclude that there are three main challenges in view reconstruction:

1. Estimation of disparity: The first main challenge in intermediate view reconstruction is the estimation of disparity field, or in other words finding the structure in the scene. As shown in Section 2.2, disparity field of a scene can be used directly to recover the depth map of the scene. The higher the quality of the depth map, the higher the quality of the reconstructed intermediate view. Unfortunately, disparity estimation is ill-posed, therefore, it is very challenging to obtain an accurate disparity field using two images.

2. Handling of occlusion areas: The second challenge in intermediate view reconstruction is handling of the occlusion areas. As we have defined in Section 3.1, an area is called an occlusion area if it is not visible in both cameras. This results from different positions of the cameras and scene structure. An example is shown in Fig. 3·3.a. Point B is visible in both cameras but points A and C are visible in either one of the cameras but not both.

There are several sub-problems related to occlusions. First, a disparity estimation method that relies on matching intensities will not be able to match point A with any point in the right image, simply because it does not exist there. Therefore, the disparity at this point cannot be defined. However, since a disparity estimation algorithm usually has no information about occlusions when starting the estimation process, it usually tries to find the best match of an occluded point with a non-occluded (visible) point in the other image, obviously a flawed approach. Two additional sub-problems arise at this point: First, is there a way to estimate occlusion areas? Secondly, even if we are able to estimate these occlusion areas, the disparity will not be defined there. How can this ambiguity be resolved?

Yet another challenge is that, as shown in Fig. 3·1, some of the texture in the scene is visible in one of the cameras and when forming the intermediate view, this information should be used to reconstruct a proper intermediate image. A reconstruction algorithm should be able to explicitly find areas that are visible only in one image and extract the texture from the original image which carries the correct information. Overall, the detection and handling of the occlusion areas belong to the most crucial steps in intermediate view reconstruction.

3. Formation of intermediate view (estimation of texture): Yet another difficulty is the formation of the intermediate view, or in order words how to estimate the texture of intermediate view. Consider that we would like to estimate the disparity of point $\mathbf{x}$ of the left image (Fig. 3·3.b). Now, consider that a disparity vector $\mathbf{d}$ is computed at $\mathbf{x}$ under constant brightness assumption, thus yielding $I_L(\mathbf{x}) = I_R(\mathbf{x}+\mathbf{d})$. If the intermediate

image is positioned at α ($\alpha = 0$ and $\alpha = 1$ correspond to positions of the left and right images respectively; $0 < \alpha < 1$ indicates a position between two input images) then we can write that $J(\mathbf{x} + \alpha\mathbf{d}) = I_L(\mathbf{x})$ or $J(\mathbf{x} + \alpha\mathbf{d}) = I_R(\mathbf{x} + \mathbf{d})$.

The main problem here is that it is unlikely that $\alpha\mathbf{d}$ will yield an integer vector representing a point on sampling grid of the intermediate view. Therefore, we can only compute intensities at *irregular* points. How can one convert intensities of the irregular points to an image or can this irregularity be avoided in the first place?

3.4 Conclusions

In this chapter, we first reviewed prior work on view reconstruction. We classified the prior work into three categories: methods that rely on (i) oversampled data, (ii) undersampled data with available geometry, and (iii) heavily undersampled data with unknown geometry. Our work falls into the third category. Next, we pointed out challenges in view reconstruction from undersampled data. Starting in the next chapter, we will provide solutions to each of these challenges and finally we will combine our solutions to achieve an occlusion-aware intermediate view reconstruction algorithm.

Chapter 4

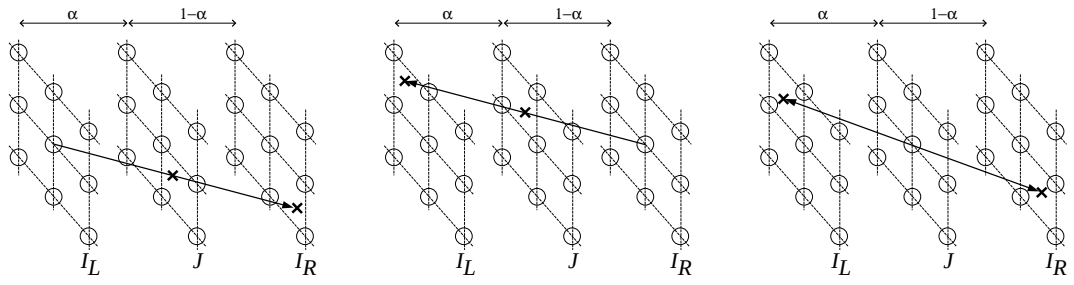
Spline-based intermediate view reconstruction

In this chapter we address one of the challenges, formation of intermediate view or texture estimation, mentioned in the previous chapter. Considering the prior work discussed, it can be seen that most reconstruction algorithms, from simple two-view disparity-compensated interpolation, to complex image-based rendering schemes, share the need to model and estimate the scene depth first. Once the scene depth is known, either explicitly or implicitly (through disparity), texture of the unknown view is estimated based on views from the real cameras and the known camera geometry.

In this chapter we focus only on the estimation of texture given a disparity field. Our motivation is to propose a better alternative to simple methods found in the literature. By extending a recently proposed method based on B-splines (Vázquez et al., 2005), we propose a view reconstruction algorithm (Ince et al., 2007a) and compare this method to widely-used pivoting based reconstruction.

4.1 Introduction

The problem of texture estimation given disparities between left and right images of a stereo pair can be better understood by examining how the disparities are computed. Let I_L and I_R be two images captured on 2-D sampling grid Λ by two closely-spaced cameras. We assume the distance between the two cameras is normalized to 1. Consider that we would like to create an intermediate view J, also defined on Λ but at a distance $0 < \alpha < 1$ from I_L, by computing disparities between I_L and I_R. Clearly, for $\alpha = 0$, we have $J = I_L$, whereas for $\alpha = 1$, we have $J = I_R$ (Fig. 4·1). It is possible to compute two vector fields: $\mathbf{d}_L$ when disparity vectors are pivoted (anchored) on the sampling grid of I_L (Fig. 4·1.a)



(a) $\mathbf{d}_L$ (b) $\mathbf{d}_R$ (c) $\mathbf{d}_J$

Figure 4·1: View reconstruction when disparity vectors are pivoted in (a) left, (b) right, and (c) intermediate image.

and $\mathbf{d}_R$ when they are pivoted on the sampling grid of I_R (Fig. 4·1.b). Under the constant-brightness assumption (Horn and Schunck, 1981), the following relationships can be written using these disparity fields:

$$I_L(\mathbf{x}) = I_R(\mathbf{x} + \mathbf{d}_L(\mathbf{x})), \quad I_R(\mathbf{x}) = I_L(\mathbf{x} + \mathbf{d}_R(\mathbf{x})), \quad \forall \mathbf{x} \in \Lambda. \tag{4.1}$$

Since brightness constancy holds along the whole disparity vector, the following relationships can be derived:

$$J(\mathbf{x} + \alpha \mathbf{d}_L(\mathbf{x})) = I_L(\mathbf{x}), \quad J(\mathbf{x} + (1-\alpha)\mathbf{d}_R(\mathbf{x})) = I_R(\mathbf{x}), \quad \forall \mathbf{x} \in \Lambda. \tag{4.2}$$

The reconstructions of $J(\mathbf{x} + \alpha \mathbf{d}_L(\mathbf{x}))$ and $J(\mathbf{x} + (1-\alpha)\mathbf{d}_R(\mathbf{x}))$ are trivial; respectively, substitute $I_L(\mathbf{x})$ or $I_R(\mathbf{x})$. However, the locations $\mathbf{x} + \alpha \mathbf{d}_L(\mathbf{x})$ and $\mathbf{x} + (1-\alpha)\mathbf{d}_R(\mathbf{x})$ usually do not belong to Λ. In fact, due to the space-variant nature of disparities, the above locations are *irregularly spaced*, whereas J defined on lattice Λ is being sought; the very problem we would like to focus on in this chapter. The two approaches typically used to solve this problem to date are described in the following sections.

4.1.1 Approach #1: Backward disparity compensation with disparity pivoting

In reconstruction based on *backward disparity compensation*, disparity vectors are defined (anchored or pivoted) on lattice Λ (i.e., sampling grid) of the view to be reconstructed

and bi-directionally point toward the known images (Mancini and Konrad, 1998; Konrad, 1999; Zhai et al., 2005). We will refer to this type of methods as *pivoting-based methods*.

As shown in Fig. 4·1.c, the disparity field $\mathbf{d}_J$ is defined on Λ in J thus forcing disparity vectors to pass through pixel positions of the intermediate view (i.e., vectors are pivoted in the intermediate view, thus the name of the method). The constant-brightness assumption now becomes:

$$I_L(\mathbf{x} - \alpha\mathbf{d}_J(\mathbf{x})) = I_R(\mathbf{x} + (1-\alpha)\mathbf{d}_J(\mathbf{x})), \quad \forall \mathbf{x} \in \Lambda. \tag{4.3}$$

Compared to equations (4.1), each lattice point of J is guaranteed to have a disparity vector and, therefore, two intensities associated with it. Although this disparity vector's end points will not necessarily intersect Λ in I_L or I_R, since intensities of both images are available on Λ, intensities off Λ can be easily calculated using spatial interpolation.

In order to reconstruct view J at a distance α, a disparity field pivoted at α is needed. This necessitates a disparity estimation for each view to be reconstructed, a significant computational burden. On the other hand, view reconstruction becomes a byproduct of disparity estimation; once left- and right-image points are selected to satisfy equation (4.3), either left or right luminance/color can be used for the intermediate-view texture. An even better reconstruction is accomplished when weighted averaging (linear interpolation) of both intensities is applied as follows (Mancini and Konrad, 1998):

$$J(\mathbf{x}) = (1-\alpha)I_L(\mathbf{x} - \alpha\mathbf{d}_J(\mathbf{x})) + \alpha I_R(\mathbf{x} + (1-\alpha)\mathbf{d}_J(\mathbf{x})), \quad \forall \mathbf{x} \in \Lambda. \tag{4.4}$$

The final step of how attributes from known images (may be more than two) are combined to recover the needed intensities is where algorithms of this type differ ; linear filtering (Franich, 1996; Mancini and Konrad, 1998; Zhai et al., 2005) and non-linear winner-take-all algorithms (Mansouri and Konrad, 2000) are some of the choices.

It is clear from (4.4) that all intermediate-view pixels are assigned an intensity and post-processing is not needed. However, in addition to the need to compute a disparity field for

each intermediate view, pivoting-based methods tend to produce somewhat blurred images due to the multiple interpolation steps involved (Mansouri and Konrad, 2000). Spatial interpolation in each view due to sub-pixel disparities (i.e., $I_L(\mathbf{x} - \alpha\mathbf{d}_J(\mathbf{x}))$, $I_R(\mathbf{x} + (1 - \alpha)\mathbf{d}_J(\mathbf{x}))$) plus interpolation between views as shown in (4.4) both induce blurring.

One should note that methods which employ ray-tracing (Gortler et al., 1996; Levoy and Hanrahan, 1996) essentially use pivoting as well. Usually, these methods apply ray-tracing at the required pixel position and find the corresponding texture in 3D model or input 2D images by using the underlying camera calibration information. Therefore, effectively, they pivot on the intermediate view to be reconstructed.

4.1.2 Approach #2: Forward disparity compensation with disparity rounding

Alternatively, in reconstruction based on *forward disparity compensation*, disparity vectors are defined on a sampling grid of a known image (or several images) while pointing, in general, to off-grid locations in the plane of the virtual image. Under constant-brightness assumption (Horn and Schunck, 1981) these locations inherit texture attributes of known images but, unfortunately, are irregularly spaced. The main issue, thus, is how to recover a regularly-spaced virtual view from these samples.

One option is to avoid the reconstruction of intermediate-view intensities off Λ by forcing the disparity-compensated locations $\mathbf{x} + \alpha\mathbf{d}_L(\mathbf{x})$ and $\mathbf{x} + (1 - \alpha)\mathbf{d}_R(\mathbf{x})$ to belong to Λ. For orthonormal lattices typically used, this means forcing $\alpha\mathbf{d}_L(\mathbf{x})$ and $(1-\alpha)\mathbf{d}_R(\mathbf{x})$ to be full-pixel vectors (Scharstein, 1996; McVeigh et al., 1996). This can be accomplished either by rounding intermediate-view positions to the nearest integer after disparity estimation (i.e., $J(nint(\mathbf{x}+\alpha\mathbf{d}_L)(\mathbf{x})) = I_L(\mathbf{x})$) or by estimating disparities under the constraint $\alpha\mathbf{d}_L(\mathbf{x}) \in \Lambda$ or $(1 - \alpha)\mathbf{d}_R(\mathbf{x}) \in \Lambda$. In consequence, most pixels in the intermediate view will have a unique intensity assigned, but some may have either no intensity or multiple intensities. Although additional post-processing using texture synthesis (to fill in the missing intensity) or depth ordering (to choose from multiple intensities) can handle such problematic areas, the resulting images are usually severely distorted. These distortions are due to disparity

rounding, that effectively implements the nearest-neighbor intensity interpolation known to cause noticeable aliasing (Keys, 1981). In case of constraining disparities during estimation, it is the coarse disparity resolution needed to meet intermediate-view lattice constraints that is the main culprit.

4.1.3 Proposed approach: Irregular to regular conversion

Recently, a solution to the problem of regularly-spaced image recovery from irregularly-spaced samples has been proposed based on spline models (Vázquez et al., 2005). We propose to adapt this approach to view reconstruction because it avoids the oversmoothing and disparity-per-view problems associated with backward disparity compensation although, admittedly, its computational complexity is higher. The main idea is based on minimization of a cost function that balances a spline-model fit to the irregularly-spaced intensity samples and spline-model smoothness. We propose an extension to this approach by *overconstraining* the solution using intensity projections from both left *and* right images.

We will evaluate the performance of the proposed reconstruction method against standard pivoting-based interpolation (backward disparity compensation). An added benefit of the spline-based reconstruction is the ability to handle occlusions more effectively. Basically, by projecting only the texture that will be visible in the intermediate view, and eliminating the occluded texture, the final view can be made free of occlusion artifacts. This will be discussed later in this dissertation.

4.2 Intermediate view reconstruction based on approximation in the space of splines

First, we will introduce B-splines, and describe a method for image reconstruction from irregularly-spaced samples using B-splines developed by Vázquez *et al* (Vázquez et al., 2005). Then, we will discuss extension of this method to view reconstruction.

4.2.1 B-splines

The computation of regularly-spaced data from irregularly-spaced input can be achieved *via* B-splines (Unser, 1999), which are commonly used as interpolation kernels. A 1D continuous function can be represented using B-splines as follows:

$$f(x) = \sum_{i=1}^{N} c_i \beta^n(x - i), \tag{4.5}$$

where c_i are spline coefficients and $\beta^n(x)$ is a B-spline of order n, defined as n-fold convolution of zeroth-order spline functions $\beta^0(x)$:

$$\beta^n = \overbrace{\beta^0(x) * \beta^0(x) * ...\beta^0(x)}^{n+1 \text{ terms}}, \quad \beta^0(x) = \begin{cases} 1, & |x| < 1/2 \\ 1/2, & x = \mp 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

A separable extension of 1D splines to multiple dimensions is trivial: $\beta(x_1, x_2, ..., x_N) = \Pi_{i=1}^{N} \beta(x_i)$. The main advantage of working with splines is that although the underlying model is a continuous representation, all computations are conducted on discrete data, specifically the spline coefficients.

4.2.2 Image reconstruction from irregularly-spaced data using cubic B-splines

Consider a set of irregular image samples, each at $\mathbf{x}_k = (x_k, y_k)$, with corresponding intensities $\widetilde{I}_k$ where $k = 1, \ldots, M$, and M is the number of samples. For compactness let us denote each pair by $P_k = \{x_k, y_k, I_k\}$, $k = 1, \ldots, M$, which will be useful in the next section.

A continuous function $f(x, y) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} c_{ij} \beta^3(x - i) \beta^3(y - i)$ is sought, where N_x and N_y are dimensions of the image, such that it best matches all samples $P = \{\mathbf{x}, \widetilde{I}\}$ by minimizing the following error:

$$D = \sum_{k=1}^{M} |f(\mathbf{x}_k) - \widetilde{I}_k|^2. \tag{4.7}$$

Although $D = 0$ can be achieved for an *interpolating* function f, this would not necessarily be the best solution because the data may contain erroneous values (local outliers). Moreover, an interpolating function could yield extreme fluctuations in areas void of input data. Therefore, a regularization term should be added to prevent such behavior. Vázquez *et al.* (Vázquez et al., 2005) proposed to apply the thin plate model as follows:

$$R = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 \right), \tag{4.8}$$

where the subscripts indicate derivatives of the continuous function f with respect to x and y. Combining the data matching and regularization terms, the cost function to be minimized is defined as $E = D + \lambda R$ where λ is the regularization factor. Minimizing E with respect to coefficients c_{ij} yields a continuous function, that, sampled on Λ, permits recovery of regularly-spaced intensities.

4.2.3 Overconstrained intermediate view reconstruction

The above spline-based reconstruction could be applied to irregularly-spaced intermediate-view intensities derived from either I_L or I_R (4.2). However, we propose to overconstrain the intermediate view by using samples derived from *both* views simultaneously as follows:

- Compute disparity fields $\mathbf{d}_L(\mathbf{x})$ and $\mathbf{d}_R(\mathbf{x})$ anchored in I_L and I_R, respectively, that satisfy (4.1).

- Create sets of irregular points $P = \{\mathbf{x}, \widetilde{I}\}$ by forward-disparity compensating intensities of I_L and I_R. In other words, create positions $\mathbf{x}_k$ and corresponding intensities $\widetilde{I}_k$ (as used in (4.7)) in the intermediate image (where $k = 1, ..., M$ and $M = N_L + N_R$; M being number total points from left (N_L) and right (N_R) images) as follows:

$$P_n^L = \{\mathbf{x}_n + \alpha \mathbf{d}_L(\mathbf{x}_n), I_L(\mathbf{x}_n)\} \text{ for } n = 1, ..., N_L, \text{ and } \mathbf{x}_n \in \Lambda_L, \tag{4.9}$$

$$P_m^R = \{\mathbf{x}_m + (1-\alpha) \mathbf{d}_R(\mathbf{x}_m), I_R(\mathbf{x}_m)\} \text{ for } m = 1, ..., N_R, \text{ and } \mathbf{x}_m \in \Lambda_R, \tag{4.10}$$

$$P = P^L \cup P^R. \tag{4.11}$$

- Find optimal function $f(\mathbf{x})$ that minimizes the cost function E (Section 4.2.2) for the set of positions and intensities P_k, $k = 1, ..., M$.
- Find values of $f(\mathbf{x})$ for all $\mathbf{x} \in \Lambda$.

The solution will be overconstrained because for an intermediate image of size $N \times M$, we use approximately $2 \times N \times M$ intensities. There is a number of benefits of overconstrained, spline-based intermediate view reconstruction. First, it allows us to use an arbitrary number of disparity fields that jointly overconstrain the solution, thus facilitating an extension to multiple views. Secondly, due to camera noise, the intensities of homologous points in I_L and I_R are rarely identical; using intensities from both images in the minimization (*via* $\widetilde{I}$) leads to a compromise solution that, in a sense, implements averaging between views (similarly to reconstruction based on disparity pivoting). Thirdly, by the very nature of spline-based reconstruction if similar intensities from both images are almost co-located, the solution will emphasize this location (sort of a weighting mechanism). Fourthly, improved image quality can be expected due to the use of original intensities in I_L and I_R, unlike in pivoting-based reconstruction that uses spatially-interpolated intensities. Fifthly, having a continuous function that represents the image can be beneficial for warping or changing scale of the image. Sixthly, the spatial regularization involved in the spline-based formulation makes the algorithm more robust to image noise than pivoting-based methods. Finally, unlike the pivoting-based method, spline-based reconstruction does not require separate disparity field for every virtual position.

4.3 Comparison of pivoting and spline-based view reconstruction

4.3.1 Comparison on ground-truth texture and ground-truth disparity

In order to compare pivoting- and spline-based view reconstruction in isolation from disparity errors, we prepared a ground-truth data set with left (I_L), right (I_R) and true midpoint (J) images. The disparity of the data set is *known*, and is free of noise and occlusion effects. This ensures a comparison of *only* reconstruction capabilities of both methods. In order

Table 4.1: Parameters used to create three ground-truth data sets (N is the downsampling factor).

Data set	N	Image size	Disparity from I_L to I_R
#1	6	350×300	(2.333,2.333)
#2	10	210×180	(1.4,1.4)
#3	15	140×120	(0.9333,0.9333)

Table 4.2: PSNR of reconstruction error for spline-based method with various regularization factors (times 10^{-3}).

	50	20	9	5	1	0.1
#1	42.04	47.46	50.62	51.11	50.52	50.20
#2	36.91	39.43	40.39	40.43	40.04	39.88
#3	31.73	32.12	31.81	31.30	29.64	28.47

to create ground-truth data set with sub-pixel disparities, we pre-filter a high-resolution image (2100×1800) to avoid aliasing after downsampling and then shift it by (7,7) and by (14,14) pixels. We downsample the original filtered image and the two filtered/shifted images by factors of $N = 6, 10$ and 15 to generate three sets of I_L, J, and I_R images. Parameters used to generate each data set are given in Table 4.1 and the left image of one set is shown in Fig. 4·2. In the following tests, a reconstruction $\widehat{J}$, at the position of the midpoint image is computed from the other two images using pivoting- and spline-based algorithms, and then average reconstruction error $J - \widehat{J}$ is calculated via PSNR (given in (3.1)).

Since the spline-based reconstruction method requires specification of the regularization factor λ, we start by evaluating its impact on PSNR of the reconstruction error. As is clear from Table 4.2, very small and very large regularization factors (on the scale of 10^{-3}) result in higher reconstruction error (lower PSNR). This is because for low regularization factors, the continuous function modeled by splines is allowed to fluctuate in areas void of input samples. On the other hand, high regularization factors inhibit fluctuations altogether thus creating oversmoothed (blurry) image.

Figure 4·2: Left image of data set #1.

Table 4.3: PSNR of the reconstruction error in absence of noise in images and error in disparities ($\lambda_1 = 50 \times 10^{-3}, \lambda_2 = 9 \times 10^{-3}$).

	Pivoting		Spline	
	Bilinear	Bicubic	λ_1	λ_2
#1	40.00	47.28	42.04	<u>50.62</u>
#2	33.37	36.50	36.91	<u>40.39</u>
#3	29.31	30.91	31.73	<u>31.81</u>

With λ calibrated, let us compare spline- and pivoting-based reconstructions under several scenarios. First, consider a noiseless case for both images and the disparities. It is clear from Table 4.3 that spline-based reconstruction performs significantly better than pivoting-based reconstruction (0.9-3.9dB improvement), if λ is carefully selected. The same test is conducted using only one set of intensities for both reconstruction algorithms, either from the left or right image (i.e., either P^L or P^R instead of P), and the results are shown in Tables 4.4.a and 4.4.b. It is clear that although the spline-based reconstruction consistently outperforms the pivoting-based reconstruction, reconstruction from either left-image intensities or right-image intensities is numerically inferior to using both images; overconstraining the solution leads to higher image quality for reasons detailed in Section 4.2.3.

In the next test, we evaluate the impact of disparity errors (e.g., resulting from disparity estimation) on reconstruction quality; uniformly-distributed white noise with range varying from [-0.1,0.1] to [-4,4] pixels is added to the ground-truth disparity field. Results shown in Table 4.5, indicate that pivoting-based reconstruction is more robust to disparity

Table 4.4: PSNR of the reconstruction error in absence of noise in images and error in disparities when reconstructing only from: (a) left image, and (b) right image ($\lambda_1 = 50 \times 10^{-3}, \lambda_2 = 9 \times 10^{-3}$), i.e., no overconstraining of intermediate view.

	Pivoting		Spline	
	Bilinear	Bicubic	λ_1	λ_2
#1	38.49	41.37	37.89	<u>42.99</u>
#2	32.45	34.40	33.71	<u>35.79</u>
#3	29.23	30.78	30.82	<u>31.83</u>

(a)

	Pivoting		Spline	
	Bilinear	Bicubic	λ_1	λ_2
#1	39.12	42.18	38.38	<u>43.88</u>
#2	33.89	36.99	35.24	<u>37.96</u>
#3	29.37	31.00	30.99	<u>32.19</u>

(b)

Table 4.5: PSNR of the reconstruction error in presence of uniformly-distributed white noise added to disparities ($\lambda_1 = 50 \times 10^{-3}, \lambda_2 = 9 \times 10^{-3}$) (Test conducted on data set #1 only.)

	Pivoting		Spline	
Range	Bilinear	Bicubic	λ_1	λ_2
$[-4, 4]$	<u>25.09</u>	24.88	24.46	22.64
$[-2, 2]$	30.19	<u>30.59</u>	29.50	27.34
$[-1, 1]$	35.20	<u>37.88</u>	35.49	34.34
$[-\frac{1}{2}, \frac{1}{2}]$	38.60	<u>43.60</u>	39.44	40.39
$[-\frac{1}{4}, \frac{1}{4}]$	39.70	<u>46.10</u>	41.21	45.36
$[-\frac{1}{10}, \frac{1}{10}]$	39.96	47.10	41.91	<u>49.27</u>

estimation errors until the error becomes negligible, after when spline-based reconstruction performs better. This behavior is not unexpected, because in pivoting-based reconstruction, each pixel is independently reconstructed using its disparity vector and therefore the errors related to disparity estimation are isolated. On the other hand, since spline-based reconstruction works on all irregular points jointly, one incorrect disparity vector affects all points in its neighborhood.

In the final test, we added white Gaussian noise with different variances to the original left and right images to better understand the impact of image noise on the reconstruction. It can be seen from Table 4.6 that spline-based reconstruction outperforms pivoting-based reconstruction, especially when regularization parameter is adjusted to the noise level.

Table 4.6: PSNR of the reconstruction error in presence of Gaussian white noise added to the original images (intensity assumed between 0 and 1, and regularization factor times 10^{-3}). (Test conducted on data set #1 only.)

	Pivoting		Spline			
σ^2	Bilinear	Bicubic	90	50	25	9
10^{-2}	25.91	24.44	26.46	25.65	24.76	23.76
$10^{-2.5}$	30.53	29.34	30.96	30.44	29.66	28.72
10^{-3}	34.59	34.21	34.69	34.86	34.46	33.70
$10^{-3.5}$	37.45	38.76	37.14	38.42	38.90	38.57
10^{-5}	39.07	42.78	38.30	40.60	42.50	43.16
$10^{-4.5}$	39.69	45.37	38.71	41.53	44.70	46.76

This can be explained again by the nature of reconstruction algorithms. Since spline-based reconstruction incorporates prior thin-plate model that can be thought of as a smoothing operator or low-pass filter, the algorithm is able to combat the noise better than pivoting-based reconstruction.

Our overall conclusion from these tests is that overconstrained spline-based reconstruction significantly outperforms pivoting-based reconstruction. Also, while spline-based reconstruction is more robust to image noise than pivoting-based reconstruction, the latter is more robust to disparity errors.

4.3.2 Comparison on ground-truth texture

In this part, we compare the two approaches on ground-truth texture data but with unknown disparities; any errors from disparity estimation affect view reconstruction performance. For ground-truth texture, we constructed several data sets using a graphics program *TrueSpace*™, which allows to create realistic 2D renderings of 3D scenes. For the 3D objects in the scene, we used VRML files courtesy of I3S Laboratory from the University of Nice at Sophia-Antipolis, France. Each data set consists of left, midpoint and right images. Again, by using left and right images, we reconstructed an intermediate image at the position of the midpoint image and calculated the reconstruction error. An optical flow algorithm with isotropic diffusion (Horn and Schunck, 1981; March, 1988) was used

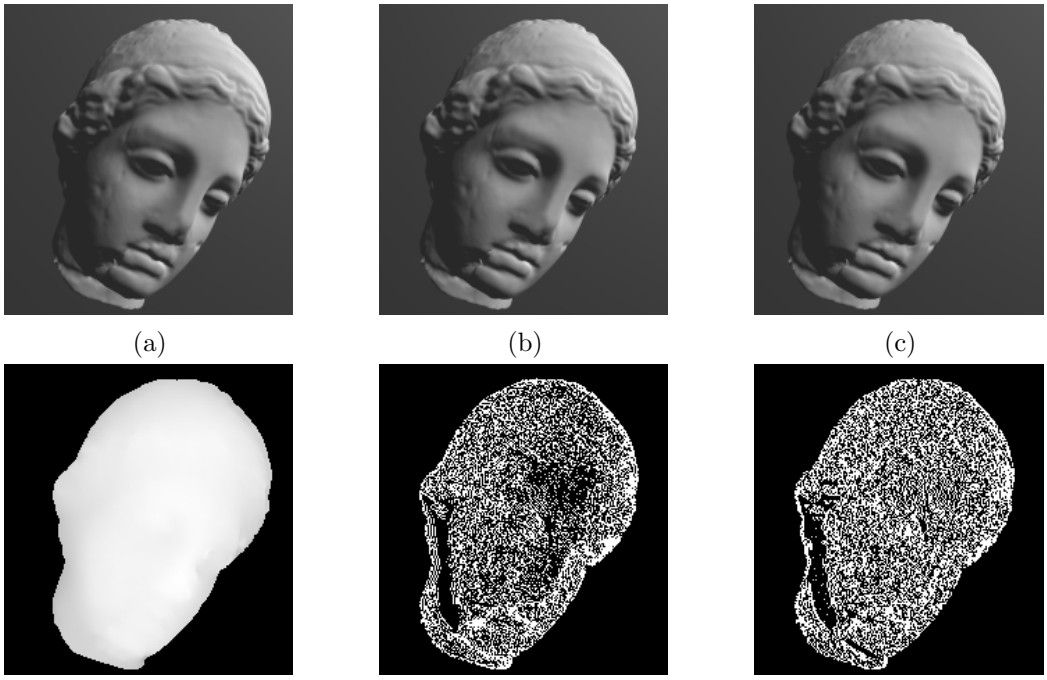


Figure 4·3: View reconstruction for parallel camera setup: original (a) left, (b) midpoint and (c) right image; (d) disparity pivoted in I_L; and locations of reconstruction error (white) greater than zero in (e) spline-based (f) pivoting-based reconstruction.

to estimate the disparities. Although for spline-based reconstruction we could have used a more advanced optical flow algorithm based on image-driven anisotropic diffusion (Ince and Konrad, 2007), it would have been unfair to the pivoting-based method as we would like to compare only the reconstruction capabilities of the methods.

Figure 4·3.a shows the midpoint image to be reconstructed. Since the data are smooth, the reconstruction quality was very good with a slight edge to the spline-based method (48.85dB) over the pivoting-based one (48.33dB). Instead of showing reconstructed images (difficult to see differences at this quality level), we are showing locations where errors are non-zero (white pixels in Figs. 4·3.e-f). It can be noticed that the number of white pixels (indicators of reconstruction error) is smaller in spline-based reconstruction. The disparity of the object (no information for background) is shown in Fig. 4·3.d.

We also tested both algorithms on the *Flowergarden* sequence, popular for its 3D qual-

ities. Although this is a monocular sequence, since the camera moves at a constant speed, while the scene is static, this setup is equivalent to multiview capture. Considering three consecutive frames (3^{rd}, 4^{th} and 5^{th}) of the sequence, we reconstructed the mid-point frame using the other two and compared the result to the original mid-point frame. The PSNR values were 29.84dB and 29.47dB for pivoting and spline-based reconstructions, respectively. Again, since the reconstruction quality was high (difficult to distinguish in print), we are giving PSNR values only.

The reconstructions presented so far have been of high quality since occluded areas were rather small. In case of significant occlusions, difficulties arise since neither pivoting-based nor spline-based method is equipped to handle them; as texture disappears between left and right images, no match can be found during disparity estimation. Consequently, incorrect disparity values are assigned to occlusion areas that, in turn, leads to poor view reconstructions. Figure 4·4 shows a stereo sequence that exhibits significant occlusions, especially near image boundaries. Part of the speaker is not visible in one of the images, moreover, there are occlusions on the picture between objects. The disparity computed by pivoting on the intermediate view and the reconstruction using this disparity field are shown in Fig. 4·4.c and d respectively. As it is clear, there are gross errors on the left boundary of the image (on the speaker, shown in closeup Fig. 4·4.f) due occlusions. Also, an area on the left-bottom of the image is not reconstructed at all, because the disparity in this area was completely incorrect. These artifacts are all due to poor disparity estimates and lack of occlusion handling. We also estimated disparity fields on the input images and then used these fields to reconstruct an intermediate image using splines. The reconstruction shown in Fig. 4·4.e (closeup in Fig. 4·4.g), although better than pivoting-based reconstruction, is not satisfactory either. Again, there are artifacts on the left boundary of the image (on the speaker and on the wall). Finally distortions on the picture between objects are shown in closeups Fig. 4·4.h-j. The arm in the left images is 'split' in intermediate images of both methods because of occlusions.

This example, once again, shows the need for accurate disparity and occlusion infor-

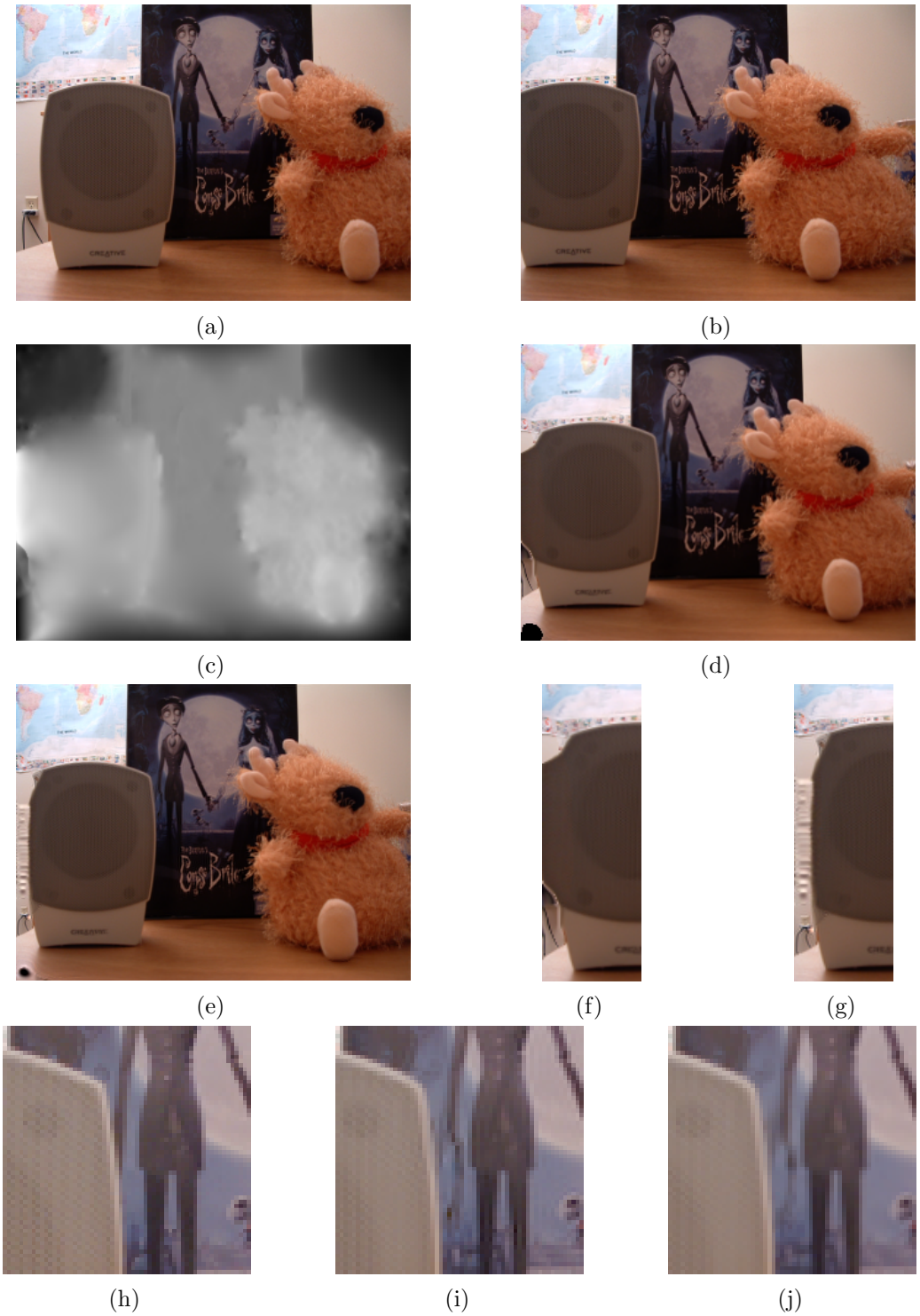


Figure 4·4: View reconstruction for a stereo sequence : (a) left (b) right image; (c) isotropically-estimated disparity pivoted in J; (d) pivoting- and (e) spline-based reconstruction; (f) closeup of (d); (g) closeup of (e); closeups of (h) left image; (i) pivoting- and (j) spline-based reconstructions.

mation, which we will focus on the rest of this dissertation. Later, in Chapter 7, we will revisit this test sequence.

Let us summarize the experimental results. Although in absence of image and disparity errors spline-based reconstruction significantly outperforms pivoting-based reconstruction, in practice, when disparity estimation introduces errors and images contain noise, the two methods have comparable performance. As for computational complexity, if a single view is to be reconstructed, pivoting-based reconstruction is less costly (single disparity field, simple averaging), but when several views need to be reconstructed, the complexity between the two methods is more comparable (spline-based reconstruction always needs two disparity estimations plus minimization).

However, a significant difference exists between the two algorithms in terms of their ability to estimate reliable disparities and to adapt to occlusions. The pivoting-based reconstruction computes disparities anchored on the sampling grid of the intermediate image that is *unknown*. This leads to oversmooth and inaccurate disparity maps. The spline-based reconstruction, however, uses disparity fields anchored in the *known* left and right images thus permitting to use underlying image to regularize the disparities. This allows to use anisotropic regularization (to be discussed in Chapter 6) which greatly improves the accuracy of disparity fields. More accurate disparity fields naturally can help us to handle occlusion areas more effectively.

Considering the problems caused by occlusions and seeing the potential of spline-based method to handle occlusion, starting from next chapter we will focus on estimation and handling of occlusion areas.

4.4 Conclusions

In this chapter, we proposed to use splines to recover intensities of the intermediate view. We first compared relative merits of pivoting-based view reconstruction and spline-based view reconstruction. We concluded that the spline-based approach outperforms the pivoting-based method except for its slightly higher sensitivity to disparity errors.

A significant computational advantage of spline-based method is that a single set of disparity fields is enough to reconstruct an intermediate view while in pivoting-based method a separate disparity field must be computed for each of the intermediate views.

Although the pivoting-based approach performs well in the absence of occlusions, the reconstruction fidelity suffers in the presence of significant occlusions. In fact, both methods fail to reconstruct high quality views in the presence of occlusions; an issue that must be addressed.

We pointed out that pivoting-based method is ineffective when handling occlusions because the disparity is anchored on the image that is being reconstructed; neither the image nor the disparity is available. On the other hand, since the spline-based method allows disparities to be computed on the grid of available views (therefore off the grid of the intermediate view), it can achieve better disparity by using the known images, and therefore can also use this reliable disparity information to extract more reliable occlusions. In the next chapter, we focus on this occlusion problem.

Chapter 5

Estimation and handling of occlusion areas

Improving the quality of reconstructed view in occlusion areas requires solving a few sub-problems. First, one needs to detect where occlusions occur. This is a problem of occlusion area detection. As mentioned in previous chapters disparity cannot be computed in occlusion areas. Therefore, second problem is to fill-in occlusion areas with correct disparity (or correct texture) that is missing. This is essentially disparity extrapolation, or we will sometimes refer to this problem as occlusion handling.

We will give a detailed analysis of problems related to occlusion areas in the next section. The remaining part of this chapter is divided into two parts. We will first discuss estimation of occlusion areas in stereo pairs and then focus on handling of occlusion areas.

5.1 Introduction

Occlusion effects in a video sequence are the result of object displacement or camera motion. What happens is that some of the texture in one frame becomes invisible in the another frame (Fig. 5.1). Occlusion effects occur in stereo images as well but this time they are due to different viewpoints of the cameras and scene structure.

By the term "occlusion area" we will refer to an area in the first image disappearing in the second image (e.g., area A in Fig. 5.1). Note that a disappearing area becomes an appearing area (also known as uncovered or newly-exposed area) and *vice versa* if the order of views is reversed , i.e., "left-to-right" versus "right-to-left".

The knowledge of occlusion and newly-exposed areas is valuable in intermediate view reconstruction for a few reasons.

The illustration in Fig. 5.1 shows, left image, I_L, right image, I_R, and the intermediate

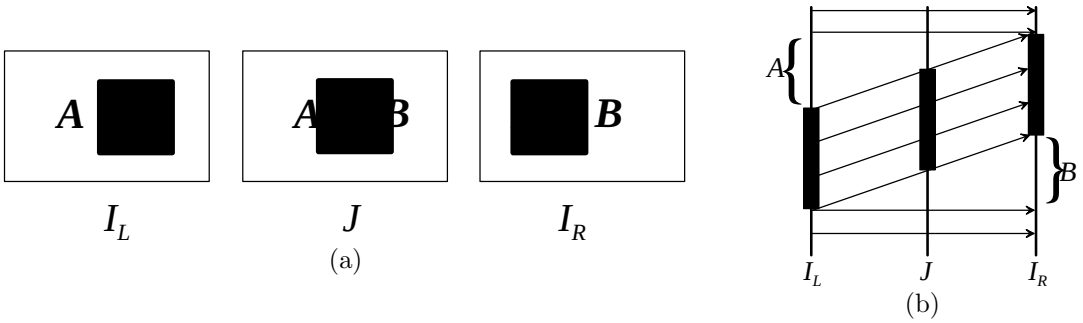


Figure 5·1: Illustration of occlusion effects in (a) two images and (b) on a horizontal cross-section of two images depicting position change of a simple object (black): Area A from I_L is being occluded in I_R by the object, while area B is being uncovered (area B would undergo occlusion had the direction of arrows been reversed).

image, J (Fig. 5.1.a) and their cross-sections (Fig. 5.1.b). Due to the displacement of the square, areas A and B are occluded/exposed between the views. Let us analyze the problems related to this occlusion effect.

1. First of all, it is important to note that disparity is undefined in occlusion areas, simply because such areas cannot be found in the other image. For example, in Fig. 5.1 points in area A of the left image I_L have no match in the right image I_R. Therefore, the disparity values in occlusion areas should not be computed so as not to yield incorrect estimates.

2. However, since disparity estimation algorithms are unaware of where occlusion areas are, they compute a disparity vector for such areas by employing some sort of regularization. These incorrect vectors will decrease the quality of disparity fields and of reconstructions that use these disparity fields.

3. Another challenge is that reconstructing texture in occlusion areas of an intermediate view is not possible without knowing where these areas are, i.e., occlusion detection problem.

4. Another implicit challenge is that even if we are able to estimate the pixels that are going to be occluded, depth (disparity) needs to be known for all pixels to be

rendered (Fig. 5.1.b). The reason is that if we do not know depth of areas A and B, then we would not know their locations in 3D space, therefore we cannot know whether they will be visible or not in intermediate image.

5. Finally, texture in the occlusion areas is visible only in one of the images. Therefore if J is to be reconstructed, the algorithm should explicitly find the correct source image. For example, partially visible area A in J is only visible in I_L and similarly partially visible areas B is only visible in I_R.

In view of all these problems, this chapter concentrates on the estimation and consequently on the handling of occlusion areas when reconstructing intermediate views. We will, first, briefly review popular methods used to estimate occlusion areas and then propose an accurate and robust, yet simple, occlusion/newly exposed area detection method (Ince and Konrad, 2005a).

In the second part of the chapter, we will focus on given occlusion areas, what can be done to recover the disparity in these areas. We will propose an image-driven disparity inpainting method for occlusion handling (Ince and Konrad, 2007) and show its better performance when compared to other methods.

5.2 Part I: Estimation of occlusion areas

Estimation of occlusion and newly-exposed areas is an inverse problem and thus is ill-posed. Occlusion detection methods proposed in the past rely on 3 or more frames (Depommier and Dubois, 1992; Chahine and Konrad, 1995; Iu, 1995; Lim et al., 2002; Ristivojević and Konrad, 2004). The idea behind these methods is that they compare intensity consistency along a trajectory formed by displacement vectors in 3 or more frames. In general, this improves reliability of occlusion estimation but requires larger buffers and is computationally more complex. More importantly, however, it requires more input data, i.e., at least three frames, that might not be always available. Considering stereo image pairs, we will focus on two-frame methods.

We start by reviewing some widely used photometric and geometric approaches to occlusion area detection.

5.2.1 Photometric approach

The usual assumption in a photometric approach is that excessive intensity matching error (disparity-compensated prediction) is observed when reference-frame pixels cannot be accurately matched in the target frame, because they disappear (Thoma and Bierling, 1989; Driessen and Biemond, 1991). This disappearance induces significant errors. If $\mathbf{d}_L$ denotes a forward disparity field anchored on the sampling grid of the left image and pointing to the right (target) image, while $\mathbf{d}_R$ denotes a backward (i.e., right-to-left) disparity field, then the corresponding disparity-compensated prediction errors at $\mathbf{x}$ are:

$$\varepsilon_{LR}(\mathbf{x}) = I_L(\mathbf{x}) - I_R(\mathbf{x} + \mathbf{d}_L(\mathbf{x})), \tag{5.1}$$

$$\varepsilon_{RL}(\mathbf{x}) = I_R(\mathbf{x}) - I_L(\mathbf{x} + \mathbf{d}_R(\mathbf{x})). \tag{5.2}$$

Typical occlusion detection methods then declare a pixel in the left image as being occluded in the right image if $|\varepsilon_{LR}| > \Theta$ for the left image, where Θ is a threshold. Similarly a point of the right image will be declared as occluded if $|\varepsilon_{RL}| > \Theta$. Note that although newly-exposed areas cannot be detected by this mechanism explicitly (pixels are not visible), effectively the occluded areas in the right image (computed using $\mathbf{d}_R$) are the newly-exposed areas for the left image.

5.2.2 Traditional geometric approach

An alternative to the photometric detection of occlusion areas is a geometric detection. Such a detection is based on the assumption that a mismatch of left-to-right and right-to-left disparity vectors is due to disappearing image areas. In particular, the following vector matching errors have been used in the past to detect occlusion and newly-exposed areas in

reference frame I_L (Proesmans et al., 1994; Izquierdo, 1997):

$$\rho_{LR}(\mathbf{x}) = \|\mathbf{d}_L(\mathbf{x}) + \mathbf{d}_R(\mathbf{x} + \mathbf{d}_L(\mathbf{x}))\|, \tag{5.3}$$

$$\rho_{RL}(\mathbf{x}) = \|\mathbf{d}_L(\mathbf{x} + \mathbf{d}_R(\mathbf{x})) + \mathbf{d}_R(\mathbf{x})\|. \tag{5.4}$$

By comparing the above errors with a threshold, decision can be made as follows:

- if $\rho_{LR} > \Delta$, then $\mathbf{x}$ in I_L is occluded in I_R or $\mathbf{x}$ in I_L is newly exposed point that was not visible in I_R,
- if $\rho_{RL} > \Delta$ then $\mathbf{x}$ in I_R is occluded in I_L or $\mathbf{x}$ in I_R is newly exposed point that was not visible in I_L.

For increased robustness, this decision can be averaged over a window, however this will sacrifice resolution of the result.

5.2.3 Ordering constraint

An *ordering constraint* assures that the order of pixels in a row of one input image will be preserved in the same line of the other image. If the order of occurrence of a point is different, then it is marked as occlusion (Geiger et al., 1995). The problem with this method is that it cannot handle thin foreground objects or narrow holes. In particular, if the disparity of an object is larger than its width, then the order will not be preserved.

This method must be applied to each row of pixels independently thus often producing uncorrelated results between rows. Finally, this method must be modified significantly, if the cameras are not parallel, because this will lead to epipolar lines that do not coincide with image scan lines.

5.2.4 Uniqueness constraint

A *uniqueness constraint* assures one-to-one mapping of pixels on corresponding rows (Marr and Poggio, 1976). In particular, a point of an image can be matched by only one point of the other image. However, this constraint fails in the presence of transparent objects. One

of the ways to implement uniqueness constraint is the traditional geometric approach as we discussed before, as it measures the match between forward and backward vector fields.

5.2.5 Other methods

Some other methods to estimate occlusion areas are:

- Continuity constraint assumes that disparity varies smoothly everywhere except object boundaries (Marr and Poggio, 1976).
- Given a disparity map, visible areas will have smaller disparity gradient, while occlusion areas will exhibit excessive gradient (Pollard et al., 1985). This is due to the smoothness constraint often used in disparity estimation methods, which assures that neighboring points have similar disparities. Since occlusion areas are near the object boundaries and a good match cannot be found in the other image, they exhibit a transition (i.e., high gradient) between neighboring objects.
- Visibility constraint (Sun et al., 2005) is a variant of the uniqueness constraint. It assures the consistency of uncovered pixels in one image with disparity of the other image, but it permits many-to-one matches in visible areas. In other words, this constraint assumes that if a pixel is newly exposed, there should not be any disparity vectors pointing to this point.

5.3 Proposed approach: A new geometric approach to the detection of occlusion areas

In the previous section, we summarized some simple methods for occlusion detection. Now we will propose a new method for the detection of occlusion and newly-exposed areas that is based on geometric properties of the disparity field. The method is applicable to any disparity field derived from an image pair. No assumption of parallel camera geometry is used; cameras can be arbitrarily arranged.

The principle of the method is based on the observation that a regular grid in the reference image plane, on which the disparity vectors are anchored, forms an irregular grid

in the target image plane after disparity compensation. Since the target image will contain no disparity-compensated projections in the newly-exposed areas, such areas can be easily detected by a simple neighborhood test. This is the basic idea behind the method; we will give more details in the next section.

It should be noted that the simple approaches discussed in Section 5.2 did not attempt to detect occlusions based on actual physical mechanism occurring during occlusions. They tried to relate mismatches in the data to occlusions. Here, we propose a new occlusion/newly-exposed area detection method based on another geometric principle of a vector field. Our approach is closer to the underlying physical model of occlusions; that is as objects displace, they leave gaps behind.

5.3.1 Detection of occlusions using the proposed method

A typical disparity field computed under some form of spatial regularization leads to converging motion vectors originating in occlusion areas of the reference frame (area A in Fig. 5·2). Such vectors cannot provide a good intensity match and assume compromise coordinates with respect to the neighboring vectors from, e.g., a moving object and static background. This convergent behavior is a compromise between the lack of intensity match and enforced spatial smoothness, and potentially leads to multiple vectors pointing to the same location in the target frame. This might suggest that a high spatial density of motion-compensated positions in the target frame (I_R in Fig. 5·2) is indicative of an occlusion area. However, in practice, it turns out that results are very sensitive to the selected density threshold. On the other hand, pixels in the target frame that did not exist in the reference frame (newly-exposed pixels in area B) have no relationship with the reference frame and, as such, cannot be pointed to by forward disparity vectors. Thus, areas in the target frame that are void of disparity-compensated projections can be relatively easily detected. This is the basis of the proposed detection algorithm.

The detection algorithm is very simple, and is equally applicable to occlusion detection if I_R is the reference frame and I_L is the target frame. Let Λ be a 2-D sampling lattice

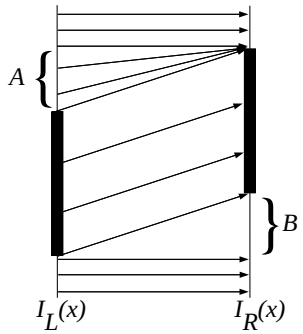


Figure 5·2: Simple occlusion process and typical disparity field; A – area to be occluded, B – area newly exposed.

for I_L and I_R limited to the domain of each image. This is unlike the standard definition of a lattice that does not constrain its extent. Also, let Γ be a set of irregular spatial positions in I_R obtained by disparity compensation of pixels from I_L, i.e., $\Gamma = \{\mathbf{y} : \mathbf{y} = \mathbf{x} + \mathbf{d}_L[\mathbf{x}], \mathbf{x} \in \Lambda\}$. Note that $card\{\Lambda\} \geq card\{\Gamma\}$ because certain points from I_L may project to the same location in I_R. Let us define an indicator function as follows:

$$\xi_i(\mathbf{x}) = \begin{cases} 1, & ||\mathbf{x} - \mathbf{z}_i|| \leq r \\ 0, & otherwise \end{cases} \qquad \mathbf{x} \in \Lambda, \mathbf{z}_i \in \Gamma.$$

For each lattice point $\mathbf{x}$ and irregular point $\mathbf{z}_i$, both in I_R, $\xi_i(\mathbf{x})$ is 1 if $\mathbf{z}_i$ is within a disk of radius r from $\mathbf{x}$. By accumulating $\xi_i(\mathbf{x})$:

$$M(\mathbf{x}) = \sum_{i=1}^{card\{\Gamma\}} \xi_i(\mathbf{x}),$$

we can measure the local density of disparity-compensated projections at each $\mathbf{x} \in \Lambda$, and by thresholding $M(\mathbf{x})$ we can find which areas of I_R exhibit the lowest density of such projections: $\mathbf{x}$ is declared newly-exposed if $M(\mathbf{x}) < \Psi$, i.e., if sufficiently few irregular points are in the vicinity of $\mathbf{x}$. We use r=2, but we test a range of values of Ψ. For areas where the motion field $\mathbf{d}_L$ is uniformly translational (regular projections), we have $M(\mathbf{x}) = 13$ for $r = 2$. At $M(\mathbf{x}) = 6$ more than half of the projections are missing suggesting vicinity of a newly-exposed area. Since it is easier to find regularly-spaced neighbors than

those spaced irregularly, the algorithm is implemented differently in practice. For each projection $\mathbf{z}_i \in \Gamma$, its neighbors $\mathbf{x} \in \Lambda$, such that $\|\mathbf{z}_i - \mathbf{x}\| \leq r$, are found, and each neighbor's counter is incremented by 1. After all $\mathbf{z}_i$ have been scanned, each counter contains the number of projections within distance of r. A computational trick that we suggest is using $r = 2\sqrt{2}$, which leads to a rectangular area of pixels rather than a circle. A clever implementation using this fact will lead to even faster computation.

5.3.2 Experimental results

Now let us show some experimental results using the new approach. In all the results shown, disparity was computed using 8×8 block matching under spatial regularization (neighboring blocks are encouraged to have similar motion vectors). The resulting vector fields are diffused on the occlusion side and have sharp boundary on the newly-exposed side of the moving object (left column of Fig. 5·3 and middle row in Fig. 5·5). In Fig. 5·3, we show results of experiment with synthetic motion of natural intensities. We measure the accuracy of detection of occluded and newly-exposed areas using symmetric difference[1] between the ground-truth pixels and the detected pixels (union of false-positives and misses), shown in the center column of Fig. 5·3 as a function of a threshold parameter for each detection method (Θ, Δ or Ψ, see Section 5.2). For each method, we show the detection result for parameter value with the lowest detection error.

Clearly, the photometric approach provides a globally-correct result that is locally very fragmented; extension of the method to a window instead of single pixel would solve this but at the cost of a significant resolution loss. The two geometric approaches result in similar occlusion/newly-exposed area descriptors, but the one based on vector mismatch leaves gaps in otherwise compact regions. In terms of the detection error the new geometric approach outperforms the traditional one by approximately 10%. As shown in Fig. 5·4, the photometric approach performs very poorly under noisy conditions. This is not unexpected

[1] Symmetric difference of two sets S_1 and S_2 is defined as $(S_1 \backslash S_2) \cup (S_2 \backslash S_1)$, equivalent to XOR boolean operator.

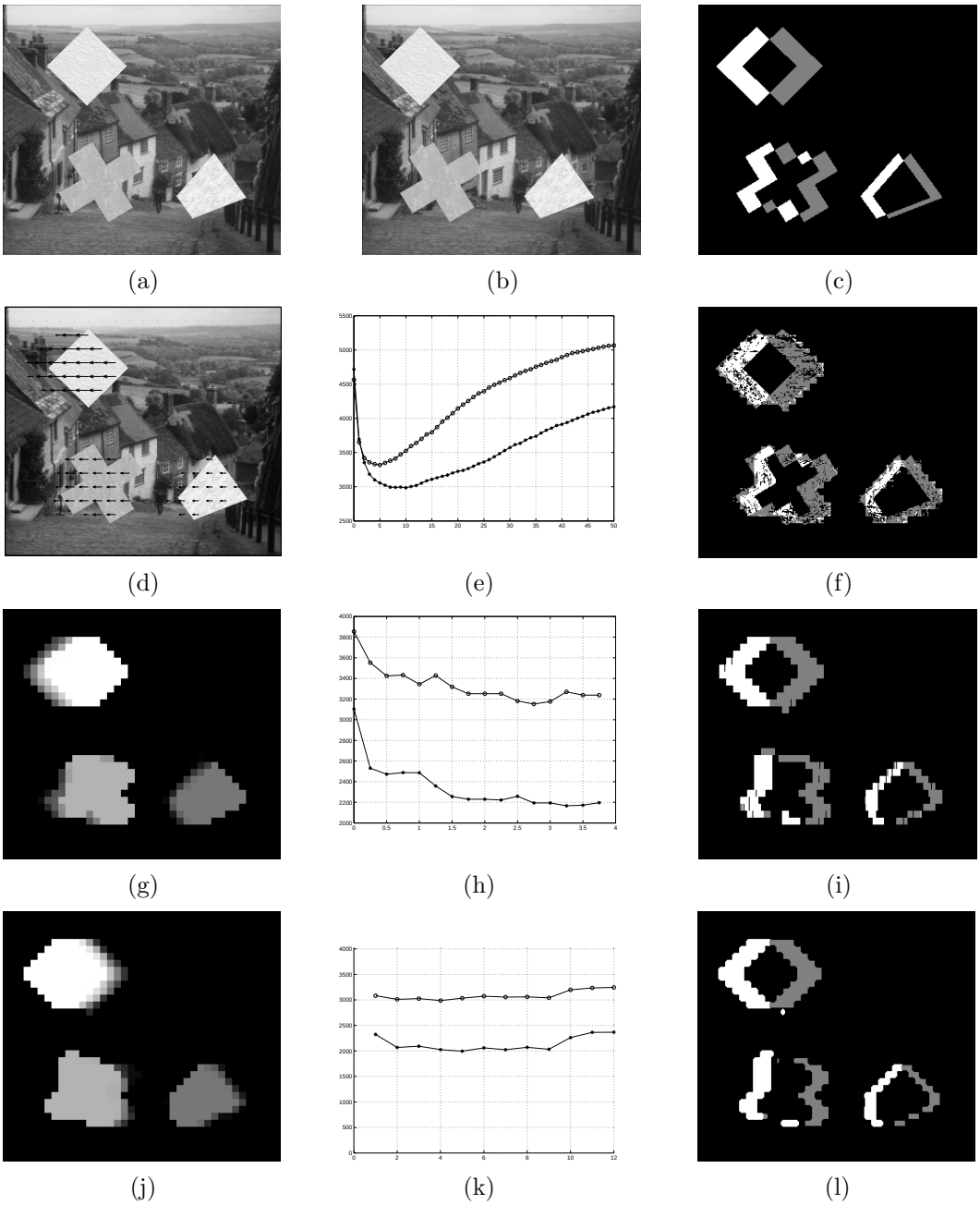
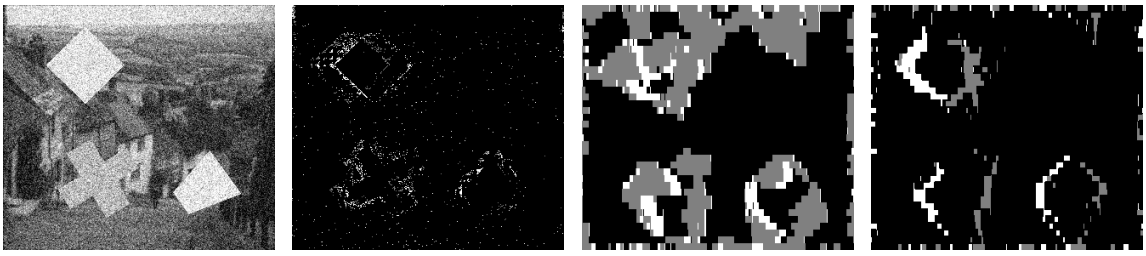


Figure 5·3: Occlusion estimation results for a synthetic sequence. In the middle column, two error plots are included, one for the detection from left to right, and the other – from right to left. In the right column, white denotes occluded area, and gray denotes newly-exposed area. (a) I_1 (b) I_2 (c) Ground-truth occlusion newly-exposed areas (d) $\mathbf{d}_L$ overlaid on I_1 (e) Photometric detect. error vs Θ (f) Photometric est. (g) $\mathbf{d}_L$ as intensity (h) Geometric detect. error vs Δ (i) Traditional geometric est. (j) $\mathbf{d}_R$ as intensity (k) New geometric detect. error vs. Ψ (l) New geometric est.



(a) I_1 + noise (b) Photometric est. (c) Traditional geometric est. (d) New geometric est.

Figure 5·4: Results for the synthetic motion sequence with additive white Gaussian noise with standard deviation σ=36 (PSNR=17.44dB).

since the detection is based directly on (noisy) intensities; using a window, again, would sacrifice resolution. The traditional geometric approach also fails in the presence of noise; disregarding the effects at image boundaries (vectors are incorrect due to the selected image boundary handling), the new method results in much more accurate estimates.

We also applied the proposed method to some well-known test sequences. As can be seen in Fig. 5·5, relatively accurate occluded and newly-exposed areas were obtained on *Flowergarden* and *Map* using this very simple, fast method. The results are not as accurate on *Tsukuba* and *Teddy* because of their relative complexity; the detected areas are in correct positions but are very fragmented. The accuracy of detection results is directly related to the quality of computed motion/disparity; better results should be possible with more sophisticated motion estimation than block-based.

5.4 Importance of the proposed occlusion detection algorithm for view reconstruction

Finally, we would like to elaborate on why the proposed method is particularly useful in intermediate view reconstruction, especially to detect the visibility of points. We mentioned this issue in Section 5.1 item #5. Let us explain how the method can be used in estimating visibility of the points.

Consider Fig. 5·6, where parts of I_L and I_R are being occluded between images. An intermediate view J is to be reconstructed. Assuming distance between I_L and I_R is normalized to 1, J is placed at α distance from I_L and at $1-\alpha$ distance from I_R. Note

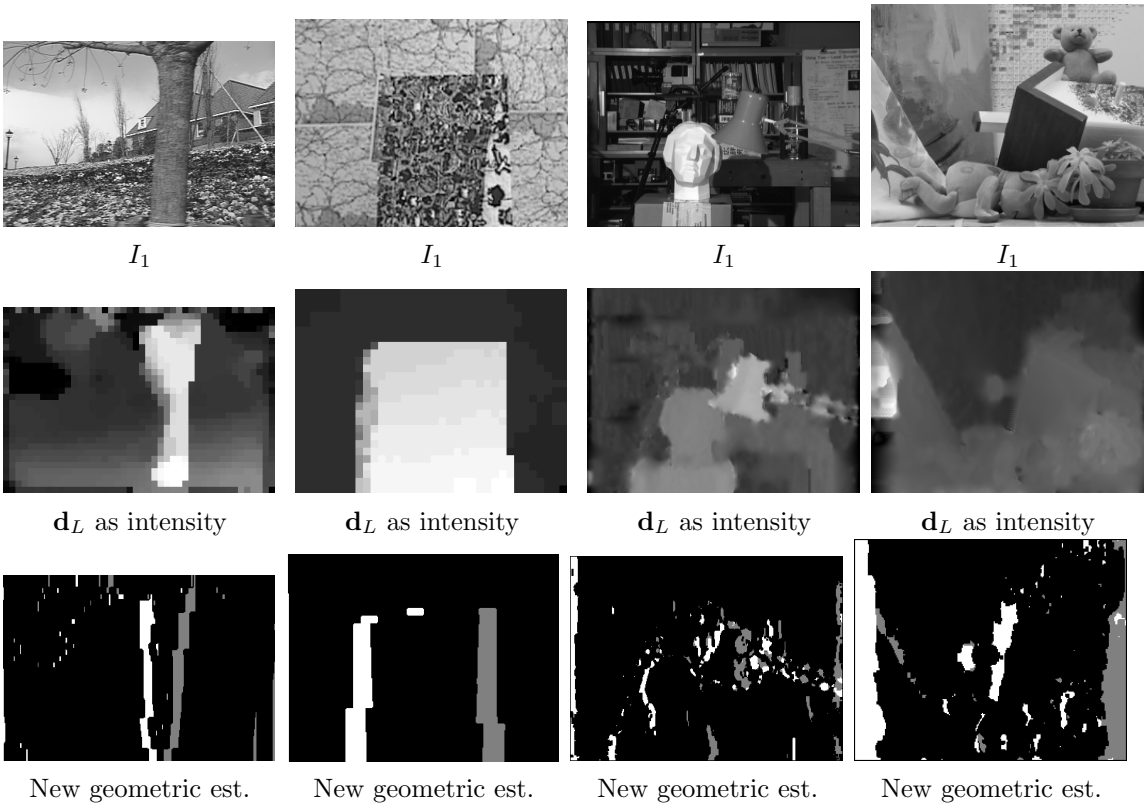


Figure 5·5: Occlusion estimation results for four well-known test sequences *Flowergarden*, *Map*, *Tsukuba* and *Teddy* (Last three test sequences are available at `http://vision.middlebury.edu/stereo/`.)

that $0 < \alpha < 1$.

Without losing generality, let us assume that the background is static. If we estimate a disparity field $\mathbf{d}_L$ pivoted on I_L toward I_R, then we can use the proposed method and estimate $O_R \cup V_R$ as shown in Fig. 5·6.a. Similarly $O_L \cup V_L$ can be estimated using $\mathbf{d}_R$ (Fig. 5·6.b). Although this information is valuable in itself, it is insufficient to estimate what parts of occlusion areas are going to be visible in the intermediate image.

Fortunately, we can adapt the method to estimate V_L and V_R as follows. Since $\mathbf{d}_L$ yields $O_L \cup V_L$, $\alpha\mathbf{d}_L$ will yield a partially exposed area which will be equivalent to V_R as shown in Fig. 5·6.c. Also, since this is an exposed area in J, it is not visible in I_L, thus must be reconstructed using texture of I_R. This way, we can also estimate which input image must be used to reconstruct the occlusion areas of J. Similarly $(1-\alpha)\mathbf{d}_R$ will be

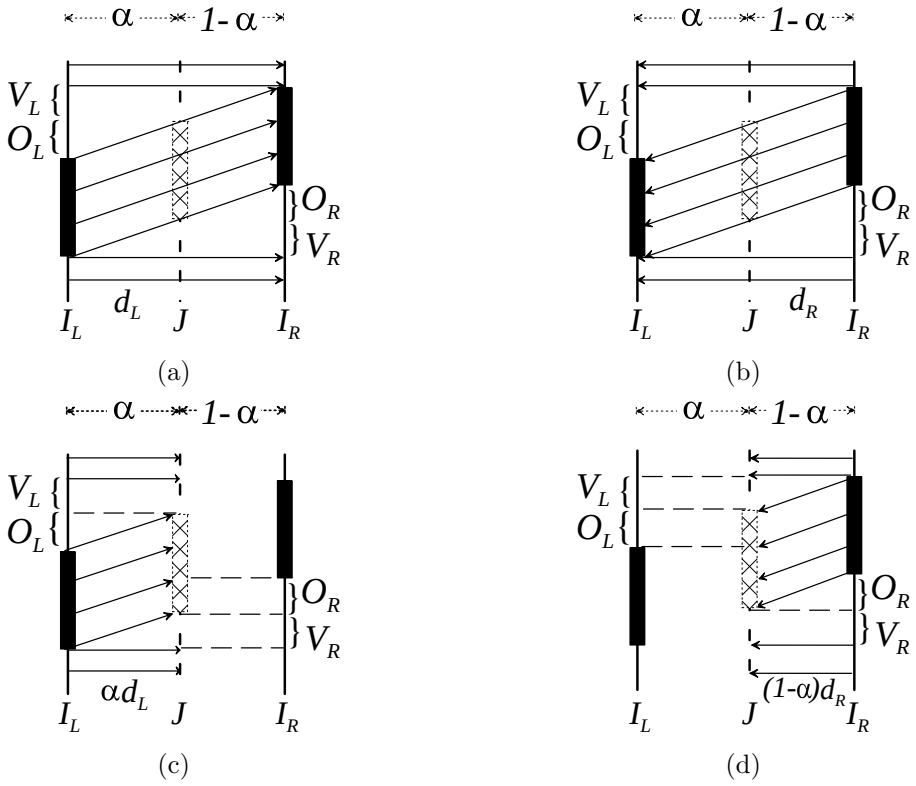


Figure 5·6: (a) $\mathbf{d}_L$ is used to estimate $O_R \cup V_R$ (b) $\mathbf{d}_R$ is used to estimate $O_R \cup V_R$ (c) $\alpha\mathbf{d}_L$ is used to estimate V_R (d) $(1-\alpha)\mathbf{d}_R$ is used to estimate V_L.

used to estimate V_L as shown in Fig. 5·6.d.

In summary, the idea here is that if full vector fields can be used to estimate large exposed areas in the known images, the vector fields multiplied by α or $1-\alpha$ will yield partially exposed areas.

Therefore, information that we can extract using this method is

(a) What part of occluded areas is visible in intermediate image?

(b) Which of the input images must be used to reconstruct occlusion areas of intermediate image?

We will later return to this topic and show experimental results by using this detection method in Chapter 7.

5.5 Part II: Handling occlusion areas: What to do in occlusion areas?

As mentioned earlier, the disparity in occlusion areas cannot be computed because texture in such areas disappears between images. This leads to difficulties when reconstructing an intermediate view. We previously mentioned this issue in Section 5.1 item #4. The basic problem is that since we have no disparity, thus no geometry information for occlusion areas, we cannot conclude whether those areas are fully occluded in intermediate view or partially visible.

Let us illustrate this problem using Fig. 5·6 again. We would like to reconstruct intermediate view J from views the I_L and I_R. In a simple scenario, first disparity between I_L and I_R would be computed, and then intensities of I_L and I_R would be disparity-compensated onto J to create the intermediate view, for example using splines as proposed in Chapter 4.

However, the area $O_L \cup V_L$ of I_L undergoes occlusion in I_R, and so does $O_R \cup V_R$ of I_R. Therefore disparity cannot be computed here. Although, V_L and V_R are visible in the intermediate view, there is no disparity information for either area, and thus rendering these areas is impossible. Therefore, in order to reconstruct these areas successfully, a proper correspondence with I_L and I_R, respectively, needs to be established through disparity.

Therefore, the problem we would like to focus on in this section is the recovery or assignment of plausible disparity values in occlusion areas, or in other words, handling of occlusion areas.

The simplest and most popular method to extrapolate disparity is to use a depth constancy assumption; disparity in a small neighborhood is assumed to be constant. For example, if there is a point $\mathbf{x}$ in the occlusion area between the points $\mathbf{x}_1$ and $\mathbf{x}_2$, a simple method to fill the area is to assume $d(\mathbf{x}) = min(d(\mathbf{x}_1), d(\mathbf{x}_2))$ for $\mathbf{x}_1 < \mathbf{x} < \mathbf{x}_2$ (McVeigh et al., 1996; Park and Inoue, 1997; Kim and Sohn, 2005). The idea behind this approach is that larger disparities indicate closer points, therefore the occlusion area should have larger depth values, i.e., small disparities. The main problem with this approach is that, such an

assumption will hold only for parallel camera structures. Also, there is no guarantee that depth in a scene is constant. The objects and background may have varying depth values.

In another, more sophisticated approach, Bertalmio *et al.* (Bertalmio et al., 2000) solved an analogous problem for images. They proposed an algorithm to fill in, or *inpaint*, missing areas in an image using surrounding intensities and their gradients. Their approach consists of two steps: first, extending available gradients into a missing area, and then applying anisotropic diffusion to propagate the available intensities into this area. The image gradients are extended first so that the subsequent anisotropic diffusion can preserve them (e.g., intensity/color discontinuities).

It is easy to notice that we would like to achieve a similar goal as Bertalmio *et al.* but with respect to disparities. Fortunately, in the case of disparity inpainting, we have a great advantage; image intensities are known in occlusion areas, and thus the first step is not needed under the assumption that image and disparity discontinuities coincide. Therefore, we propose to extrapolate disparities using anisotropic diffusion driven by image gradient as will be discussed next.

5.6 Proposed approach: Image-driven disparity inpainting

Let $\mathbf{x} = (x, y)^T \in \Omega$ be a spatial position in image I defined on Ω. Also, let $\{\mathbf{d}(\mathbf{x})\}_{\mathbf{x}\in\Omega}$ be a disparity field to be computed; $\mathbf{d} = [u, v]^T$ with u and v being, respectively, horizontal and vertical components of the disparity vector $\mathbf{d}$. Finally, let $O \subset \Omega$ be an occlusion area in image I. To inpaint disparities, we exploit the underlying image structure by minimizing the following cost function with respect to $u(\mathbf{x}), \mathbf{x} \in O$:

$$F_{\mathbf{x}}(u, I) = \nabla^T u(\mathbf{x}) \begin{bmatrix} g(|I^x(\mathbf{x})|) & 0 \\ 0 & g(|I^y(\mathbf{x})|) \end{bmatrix} \nabla u(\mathbf{x}), \tag{5.5}$$

where $g(\cdot)$ is a monotonically-decreasing function and I^x, I^y are horizontal and vertical derivatives of I, respectively. A similar cost function, $F_{\mathbf{x}}(v, I)$, can be written for the

vertical component $v(\mathbf{x})$ as follows:

$$F_{\mathbf{x}}(v, I) = \nabla^T v(\mathbf{x}) \begin{bmatrix} g(|I^x(\mathbf{x})|) & 0 \\ 0 & g(|I^y(\mathbf{x})|) \end{bmatrix} \nabla v(\mathbf{x}). \tag{5.6}$$

Therefore, in order to inpaint both components of $\mathbf{d}(\mathbf{x})$ we minimize $F_{\mathbf{x}}(u, I) + F_{\mathbf{x}}(v, I)$.

This leads to anisotropic diffusion; disparities in occlusion areas are diffused while accounting for the underlying image gradient. Assuming that the gradient magnitude within an object is small, an iterative algorithm minimizing (5.5,5.6) will diffuse disparities inside each object only. The edge-stopping function $g(\cdot)$ will prevent diffusion across object boundaries because gradient is usually large there.

The cost function (5.5) is very similar to the one proposed by Perona and Malik (Perona and Malik, 1990). However, while in Perona and Malik's case image intensity undergoes smoothing and at the same time drives the edge-stopping function $g(\cdot)$ to control anisotropy during diffusion, in our case horizontal and vertical disparity components are being (separately) smoothed but anisotropy is controlled by the underlying image intensity. This approach has also been used by others to regularize disparities in optical flow based disparity estimation methods (Kim and Sohn, 2005; Huang and Dubois, 2005).

5.6.1 Experimental results

In Fig. 5·7, we compare results of the proposed image-driven anisotropic disparity diffusion with those of other extrapolation approaches on a synthetic image. We assume that we are given a stereo pair (left image shown in Fig. 5·7.a) and a partially estimated horizontal (1D) disparity map with occlusions marked in black (Fig. 5·7.b). We would like to extrapolate the disparity map in occlusion area closely approaching the ground-truth disparity shown in Fig. 5·7.c as an intensity image and in Fig. 5·7.d as a 3D surface.

A simple extrapolation using depth constancy along epipolar line (Kim and Sohn, 2005) leads to a patch-like result shown in Fig. 5·7.e, because depth constancy approach does not hold here. On the other hand, isotropic diffusion where $g(x) = 1$ (Fig. 5·7.f) is overly

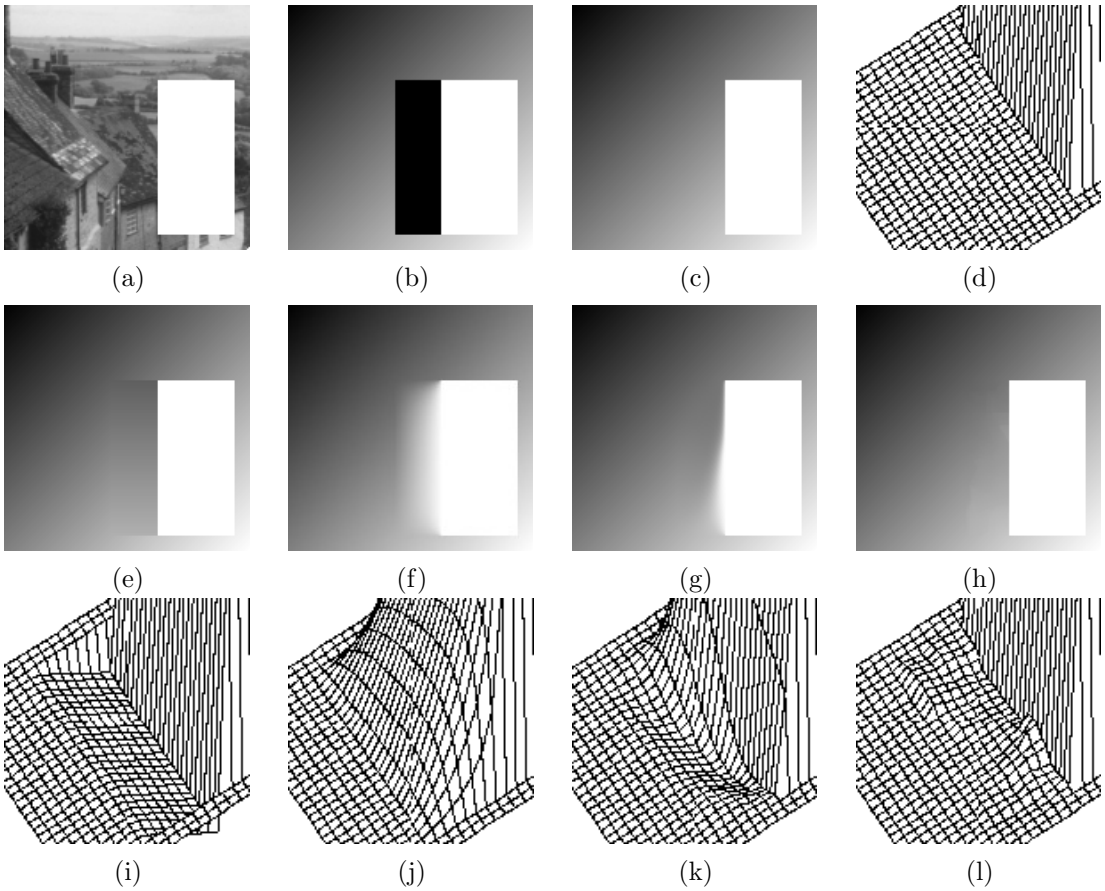


Figure 5·7: Comparison of disparity extrapolation methods on synthetic images: (a) I_L; (b) partial disparity map with ground-truth occlusions (black); ground-truth disparity as (c) intensity image and (d) 3D surface; and the extrapolated disparity based on (e) depth constancy along epipolar line; (f) isotropic diffusion; (g) standard inpainting; (h) proposed approach; (i-l) corresponding 3D surfaces of disparity extrapolated in occlusion areas.

smooth, because it disregards the underlying image gradient and diffuses into occlusion area from all sides. The results of standard inpainting method of Bertalmio *et al.* and the proposed approach are shown in Figs. 5·7.g and 5·7.h, respectively. The last row in Fig. 5·7 shows the same results as a 3D surface around occlusion area.

Although standard inpainting preserves structure much better than depth constancy and isotropic diffusion, there is still unwanted smoothing especially at the bottom part of the white rectangle. This is due to a weaker gradient in the image around that area. Standard inpainting fails to recognize that area as an edge because it does not use under-

lying image. The image-driven anisotropic diffusion, however, produces an extrapolated disparity field with a clear discontinuity and is barely distinguishable from ground-truth.

5.7 Can we jointly estimate and handle occlusion areas?

Above, we showed the efficacy of image-driven disparity inpainting on a synthetic image. Although it works very well, this approach stands out as a post-processing step. If there were errors in occlusion estimation step, then these errors would not be corrected in the occlusion handling stage. Even worse, if there were errors in the disparity estimation stage, estimated occlusion areas would be erroneous as well.

Clearly, step-by-step approach will have limited capabilities. The question that we would like to further elaborate is whether it is possible to jointly solve both problems in a single formulation that will lead to better results. In the next chapter, we will focus on this problem.

5.8 Conclusions

This chapter first concentrated on how to estimate occlusion areas. We proposed a simple yet effective way of estimating occlusion and newly exposed areas. We pointed out why this method is especially important in view reconstruction as it is able to find areas that will be occluded in the intermediate view. In the second part, we proposed an image-driven disparity inpainting method to assign disparities to occlusion areas. The experimental results demonstrate better performance compared to other methods. Finally, we discussed why a step-by-step approach is inefficient in disparity and occlusion estimation. In the next chapter, we will utilize image-driven inpainting in an optical flow framework that will jointly solve both problems.

Chapter 6

Occlusion-aware optical flow estimation

In this chapter, we propose an optical flow method which estimates disparities, implicitly estimates occlusion areas and assigns plausible disparities to occlusion areas. We will start by introducing optical flow work of Horn and Schunck (Horn and Schunck, 1981) and review various improvements to this method. Then, we will introduce our occlusion-aware optical flow algorithm (Ince and Konrad, 2007).

6.1 Introduction and motivation for a joint formulation

As we pointed out in the previous chapter, disparity[1] is undefined in occlusion areas and, when only two images are available, a usual remedy is to extrapolate (inpaint) optical flow in occlusion areas. Ideally, one would identify occlusion areas in advance and use the detected labels in an occlusion-adaptive optical flow estimation, relying solely on spatial regularization to fill-in the occlusion areas. However, it is unclear how to find occlusion areas without first computing optical flow. Hence, in practice, occlusion-unaware optical flow is computed first, resulting in incorrect vectors in occlusion areas; a best-effort match is forced despite the fact that a feature being matched is absent from the other image. Then, occlusion areas are identified, and, finally, the optical flow is corrected in these areas (Kim and Sohn, 2005). This three-step approach is often ineffective since incorrect vectors from occlusion areas had already affected their neighbors in visible areas due to spatial regularization typically used. Moreover, this approach does not bootstrap optical flow estimates to improve occlusion detection results.

[1] In this chapter, displacement of points between images are called disparity or motion or more generally optical flow depending on the context.

Our motivation is to recover dense disparity (depth) from *two* images *everywhere* in the domain of either image and also to minimize the impact of occlusion areas on disparity estimates at visible points. We deal with deficiencies of the three-step approach by proposing a variational formulation that jointly estimates disparity vectors, implicitly detects occlusions and extrapolates disparities in occlusion areas.

The basic idea behind the method is that the evolving occlusions force disparity extrapolation (*via* diffusion) by automatically disabling an intensity matching term at occluded pixels, but permit standard disparity estimation at visible pixels. By using anisotropic diffusion driven by image gradient, the interaction between occlusion-area and visible-area disparity vectors is inhibited. At the same time, this joint formulation solved iteratively facilitates interaction between disparity vectors and occlusion labels, thus leading to more coherent solutions.

Let us first start by introducing optical-flow estimation.

6.2 Optical-flow-based disparity estimation

As shown in Section 2.2, disparity field of a scene can be used directly to recover the depth map of the scene. The higher the quality of the depth map, the better will be the reconstructed intermediate view. Unfortunately, since disparity estimation is ill-posed, it is very difficult to extract reliable disparity fields using only two images.

Among many disparity estimation methods (see (Scharstein and Szeliski, 2002) for an overview) we would like to focus on optical flow methods due to their accuracy.

Optical-flow-based motion estimation was introduced by Horn and Schunck in their seminal paper (Horn and Schunck, 1981) and was later studied by March (March, 1988) in the context of disparity estimation. The main constraint in motion and disparity estimation is the constant brightness assumption which assumes that intensity of point does not change

between images. This constraint leads to the following minimization[2]:

$$\arg\min_{u(\mathbf{x}),v(\mathbf{x})} \iint_{\Omega_L} \left(I_L(x,y) - I_R(x+u(\mathbf{x}), y+v(\mathbf{x}))\right)^2 d\mathbf{x}. \tag{6.1}$$

where $u(\mathbf{x})$ and $v(\mathbf{x})$ are velocities (equivalent to displacement when unit time step is used) that are sought in horizontal and vertical directions.

The second constraint proposed by Horn and Schunck is the smoothness constraint which states that points in a small neighborhood usually have similar velocities. Alternatively, this means that sudden changes in velocity fields are unlikely, which can be written as follows:

$$\arg\min_{u(\mathbf{x}),v(\mathbf{x})} \iint_{\Omega_L} |\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2 d\mathbf{x}. \tag{6.2}$$

By combining these two constraints using a smoothness (or regularization) factor λ, we obtain

$$\arg\min_{u(\mathbf{x}),v(\mathbf{x})} \iint_{\Omega_L} \left(I_L(x,y) - I_R(x+u(\mathbf{x}), y+v(\mathbf{x}))\right)^2 d\mathbf{x} + \lambda \iint_{\Omega_L} |\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2 d\mathbf{x}. \tag{6.3}$$

The final iterative solution can be found in (Horn and Schunck, 1981). The smoothness factor λ controls the relative importance of both terms. If λ is increased, variations in the displacement field will be penalized, therefore a smoother field will be obtained.

We focus on optical flow methods because they yield a dense vector field which means that every point in the image has a distinct vector as opposed to, for example, block matching which yields a piecewise-constant field. Secondly, optical flow methods are able to generate floating-point (i.e., sub-pixel) disparity vectors, that can highly increase picture quality in some applications. Yet another reason is their ability to generate two dimensional vectors fields, unlike methods such as dynamic programming. Stereo images

[2]We should note that Horn-Schunck originally derived optical flow equation by expanding the constant brightness assumption $dI/dt = 0$ as follows: $\partial I/\partial x \cdot \partial x/\partial t + \partial I/\partial y \cdot \partial y/\partial t + \partial I/\partial t = 0$. The equation in (6.1) is valid, but more related to the approach of March.

will always have a vertical disparity component unless images are captured by perfectly parallel cameras.

As we mentioned, many reliable disparity estimation methods fail in presence of vertical disparity. Although, the parallel camera setup has been extensively studied, even a slight rotation of cameras, e.g., toed-in structure, creates two dimensional disparity as well as keystone distortions. Therefore, algorithms designed for parallel camera setup usually preprocess images using a method called rectification (Papadimitriou and Dennis, 1996) so that images are aligned in order to eliminate vertical disparity. This kind of setup, however, poses two problems. First, rectification involves resampling of the original images, which decreases image quality. The resampling process uses interpolation which is equivalent to low pass filtering and causes blur. The second problem is that the algorithms require extra computation time for rectification and need to access internal camera parameters, such as focal length. By using optical flow methods, we avoid both of these problems and we can work on any pair of images without any need to access calibration parameters. Moreover, the ability to capture vertical disparities will also help when we would like to apply our algorithm to monoscopic video sequences to capture motion instead of disparities in order to generate intermediate frames.

6.2.1 Prior improvements to optical flow methods

One of the main problems with the Horn-Schunck formulation is that the regularization term disregards object boundaries. The main assumption of Horn and Schunck for the regularization term was that objects in real world are rigid, and, therefore, neighboring points in an image should have similar velocities. Although being a reasonable assumption, it fails in the presence of object boundaries. Therefore, two neighboring points in an image, one of them belonging to an object and the other to the background, will be forced to have similar velocities. This kind of regularization is called *isotropic regularization* (stemming from equation (6.2)). Since the objects and background have different displacements, an ideal vector field should exhibit a discontinuity at object boundaries. Unfortunately, the

result of isotropic diffusion is that object boundaries, which are semantically important cues, are not preserved in vector fields. What happens is a gradual change from object displacement towards background displacement.

Methods have been proposed to deal with such smoothing by assuming that intensity discontinuities coincide with object boundaries, such as image-adaptive, isotropic diffusion (Alvarez et al., 1999). In this case the smoothing term is multiplied by the inverse of image gradient at that point. In case of a strong gradient, i.e., an edge, the smoothness term is disabled.

Others have used image-adaptive anisotropic diffusion (Nagel and Enkelmann, 1986; Mansouri et al., 1998; Alvarez et al., 2002b; Kim and Sohn, 2005; Huang and Dubois, 2005). These methods replace the isotropic regularization term $\nabla^T d(\mathbf{x}) \nabla d(\mathbf{x})$ in equation (6.2) with $\nabla^T d(\mathbf{x}) D(\nabla I) \nabla d(\mathbf{x})$ where $D(\nabla I)$ is a projection matrix defined, for example, as follows (Alvarez et al., 2002b) (other Ds are possible):

$$D(\nabla I) = \frac{1}{2\lambda + |\nabla I|^2} \begin{bmatrix} I_x^2 & -I_x I_y \\ -I_x I_y & I_y^2 \end{bmatrix} + \lambda^2 \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \tag{6.4}$$

where K is a scalar.

It is also possible to control smoothness by using the estimated vector field to determine the gradient information. For example, if there happens to be a high gradient area in the vector field, this will be enhanced as the estimation progresses. We refer the reader to a paper of Weickert and Schnörr (Weickert and Schnörr, 2001) for a detailed taxonomy of regularizers.

In another type of approach it is possible to replace the quadratic function of the regularization term with non-quadratic robust smoothness terms such as the Lorentzian (Black and Anandan, 1996; Robert and Deriche, 1996). These types of methods do not rely on image gradient but on robust properties of non-quadratic regularizers.

These improved optical flow methods lead to discontinuity-preserving vector fields, as shown in Fig. 6·1, but still produce erroneous results at object boundaries since incorrect

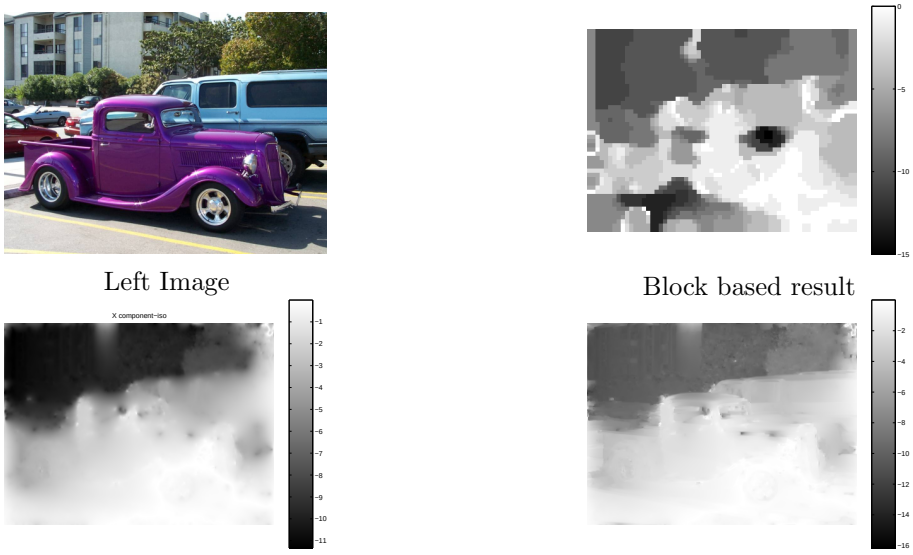


Figure 6·1: Results of disparity estimation of Truck (only horizontal disparity is shown). Stereo pair is from `http://www.stereovision.net`.

intensity matches are allowed despite occlusions. As mentioned earlier, the problem could be corrected to a degree in three steps (occlusion-ignorant optical flow estimation, detection of occlusions, and correction of the flow), however such a procedure is cumbersome and there is no interaction between visible-area flow vectors, occluded-area flow vectors and occlusion labels. Therefore, of interest are methods that would perform optical-flow estimation, occlusion detection, and optical flow extrapolation jointly.

Since occlusion models are inherently discrete in amplitude (image point is occluded, newly-exposed or visible), their representation within the variational framework is usually implicit, for example by means of a continuous-amplitude inconsistency field (Proesmans et al., 1994). Using such a representation, Proesmans *et al.* (Proesmans et al., 1994) proposed an edge-preserving optical flow estimation with forward/backward vector differences (geometric constraint) serving as inconsistency field. This field was used to control the local strength of image-adaptive, isotropic diffusion. A similar idea was also proposed by Alvarez *et al.* (Alvarez et al., 2002a); forward/backward vector field consistency was enforced *via* an energy term in variational framework, while each vector field was regularized

using anisotropic diffusion. In both works, occlusions were detected by thresholding the final disparity field (geometric) mismatch.

Strecha and Van Gool (Strecha and Gool, 2002) introduced the idea of disabling a data-matching energy term in optical flow formulation in order to select the best prediction among camera pairs in a multiple-camera system. Since weights used for disabling exploit a geometric constraint, this approach can be considered occlusion-adaptive. Although similar disabling idea is used in our approach, the rest of our formulation is quite different. More importantly, however, our approach uses only two images and cannot seek optimal prediction from other images.

Sun *et al.* (Sun et al., 2005) also proposed joint disparity estimation and occlusion detection using a visibility constraint but treated the disparity correction in occlusion areas as a post-processing step. Moreover, this corrective step assumed that depth in occlusion areas and surrounding pixels is constant on epipolar line. Although often valid, this assumption fails for image backgrounds with varying depth. The discrete disparity prior used was equivalent to anisotropic diffusion in variational formulations, while the overall discrete formulation was solved using belief propagation. Lim *et al.* (Lim et al., 2002) also proposed to jointly estimate bidirectionally-consistent (forward/backward) motion fields and occlusion labels from a pair of images using Markov random fields in a Bayesian framework. However, the occlusion detection mechanism relied on a photometric constraint and no vector correction was performed in occlusion areas. Graph cut methods (Kolmogorov and Zabih, 2001; Xiao and Shah, 2005) have been used in disparity/motion estimation under occlusions as well. These methods explicitly define an occlusion term in the formulation, however intensity matching term and occlusion term are usually not directly coupled unlike in the approach proposed here.

Very recently, Xiao *et al.* (Xiao et al., 2006) proposed a similar approach to ours, that implemented in a loop, can be considered a joint approach. Although occlusion detection is embedded into the formulation like in our method, there are important differences as well. First, Xiao *et al.* employ a bilateral filter and locally adjust filter strength by using

precomputed occlusion labels in each iteration, while we use anisotropic diffusion and let the joint formulation drive the diffusion process by using "soft" occlusion information (no explicit occlusion labeling step). Secondly, the occlusion detection uses a photometric constraint, that is unreliable under image noise and illumination changes (Ince and Konrad, 2005a), while we use a geometric constraint (Section 5.2.2).

In view of the prior work, our approach, resembling the approach of Xiao *et al.* (Xiao et al., 2006), makes an important contribution. Namely, we propose a joint variational formulation for disparity estimation, (implicit) occlusion detection, and disparity extrapolation. Benefits of our new formulation are threefold. First, the image-driven anisotropic diffusion fills-in disparities in occlusion areas respecting image structure (intensity discontinuities), therefore providing plausible solutions (unlike constant-disparity extrapolation along epipolar line). Secondly, the joint formulation permits interaction between disparities and occlusions during estimation, thus allowing mutual corrections (unlike in the case of three-step approaches). Thirdly, by disabling the data-matching term in occlusion areas, the estimation bias of disparity vectors is eliminated since the system relies exclusively on anisotropic diffusion there (in visible areas data-driven flow and diffusion work together).

6.3 Proposed approach: Joint disparity estimation/inpainting

As shown in Section 5.6, image-driven anisotropic disparity diffusion can be an effective tool in extrapolation of disparities in occlusion areas. However, we assumed there that the disparity field and occlusion map are known, and thus disparity inpainting is basically a post-processing step. Now, we propose a new approach that combines disparity estimation, occlusion detection and disparity extrapolation in a single formulation.

Let $I_L : \Omega_L \rightarrow R^+$, $I_R : \Omega_R \rightarrow R^+$, and let $\mathbf{x}$ belong either to Ω_L or Ω_R. We would like to compute two disparity fields:

$$\{\mathbf{d}_L(\mathbf{x}) = [u_L(\mathbf{x}), v_L(\mathbf{x})]^T\}_{\mathbf{x}\in\Omega_L}, \tag{6.5}$$

$$\{\mathbf{d}_R(\mathbf{x}) = [u_R(\mathbf{x}), v_R(\mathbf{x})]^T\}_{\mathbf{x}\in\Omega_R} \tag{6.6}$$

that, for pixels visible in both images, minimize some metric of the following photometric errors:

$$\begin{aligned} \rho_{LR}(\mathbf{x}) &= I_L(\mathbf{x}) - I_R(\mathbf{x} + \mathbf{d}_L(\mathbf{x})), \\ \rho_{RL}(\mathbf{x}) &= I_R(\mathbf{x}) - I_L(\mathbf{x} + \mathbf{d}_R(\mathbf{x})). \end{aligned} \tag{6.7}$$

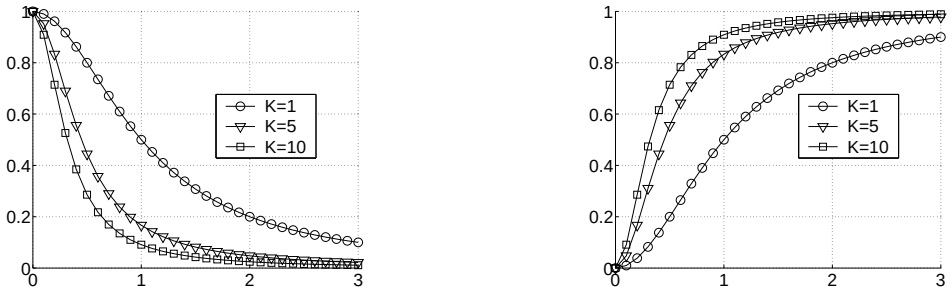
However, to distinguish occluded and visible image areas, we propose to use the disparity mismatch (geometric constraint):

$$\begin{aligned} \epsilon_L(\mathbf{x}) = &\quad \|\mathbf{d}_L(\mathbf{x}) + \mathbf{d}_R(\mathbf{x} + \mathbf{d}_L(\mathbf{x}))\|, \\ \epsilon_R(\mathbf{x}) = &\quad \|\mathbf{d}_R(\mathbf{x}) + \mathbf{d}_L(\mathbf{x} + \mathbf{d}_R(\mathbf{x}))\|, \end{aligned} \tag{6.8}$$

where $\|\cdot\|$ denotes Euclidean norm. Both $\epsilon_L(\mathbf{x})$ and $\epsilon_R(\mathbf{x})$ are expected to be small for visible pixels and larger for occluded pixels. Note that although photometric errors (6.7) could have been used as occlusion detectors as well, they are less robust to noise and intensity variations (Ince and Konrad, 2005a).

In order to model data-matching, disparity and occlusion constraints, we propose three pairs of energy functions that combined together will lead to the final formulation.

Photometric constraint: Since photometric constraints, expressed through errors (6.7), do not hold in occlusion areas (errors are large), we need to disable their impact on the overall cost function whenever ϵ_L or ϵ_R is large. We can achieve this by multiplying magnitude (or square) of the photometric error by a weight function inversely proportional to the disparity mismatch; the larger the mismatch, the smaller the contribution of this photometric error to the overall cost function. We propose a monotonically decreasing weight function $D(z) = 1/(1 + Kz^2)$, with constant $K > 0$ controlling function's slope (Fig. 6·2). It is possible to use other functions such as a Gaussian function for D. As shown in Fig. 6·2, $D(\epsilon_L(\mathbf{x}))$ and $D(\epsilon_R(\mathbf{x}))$ approach zero as the disparities $\mathbf{d}_L, \mathbf{d}_R$ are less and less capable of compensating each other. We define the first pair of energy functions



(a) (b)

Figure 6·2: Weights: (a) $D(z)$ and (b) $1 - D(z)$ for various values of K.

(photometric) as follows:

$$\begin{aligned} E_L^P &= \iint_{\Omega_L} D(\epsilon_L(\mathbf{x}))[\rho_{LR}(\mathbf{x})]^2 d\mathbf{x}, \\ E_R^P &= \iint_{\Omega_R} D(\epsilon_R(\mathbf{x}))[\rho_{RL}(\mathbf{x})]^2 d\mathbf{x}. \end{aligned} \tag{6.9}$$

These energies differ from the usual optical flow formulation by their ability to disable the impact of photometric error when the disparities do not compensate each other. This is essential because these areas are most likely occluded and the intensity matching term is not beneficial. On the contrary, it may lead to false solutions.

Inpainting (diffusion) term: We embed the idea of image-driven disparity extrapolation through the second pair of energy functions:

$$\begin{aligned} E_L^S &= \iint_{\Omega_L} (F_{\mathbf{x}}(u_L, I_L) + F_{\mathbf{x}}(v_L, I_L)) d\mathbf{x}, \\ E_R^S &= \iint_{\Omega_R} (F_{\mathbf{x}}(u_R, I_R) + F_{\mathbf{x}}(v_R, I_R)) d\mathbf{x}, \end{aligned} \tag{6.10}$$

with $F_{\mathbf{x}}$ defined in (5.5). Note that energies (6.9) and (6.10) jointly lead to edge-preserving regularization (no disparity smoothing across strong intensity gradients) when $D(\cdot)$ is close to 1, but result in disparity inpainting when $D(\cdot)$ is around zero (i.e., possible occlusion area), since the data-matching terms are disabled.

Occlusion prior term: The energies (6.9) can be easily made arbitrarily small by

choosing vector fields with sufficiently large ϵ_L and ϵ_R (6.8) for all $\mathbf{x}$. In order to prevent this, we propose an explicit occlusion model through the following energies:

$$\begin{aligned} E_L^O &= \iint_{\Omega_L} (1 - D(\epsilon_L(\mathbf{x})))d\mathbf{x}, \\ E_R^O &= \iint_{\Omega_R} (1 - D(\epsilon_R(\mathbf{x})))d\mathbf{x}. \end{aligned} \tag{6.11}$$

Note that $1 - D(\epsilon_L(\mathbf{x}))$ and $1 - D(\epsilon_R(\mathbf{x}))$ approach 1 as the disparity mismatches $\epsilon_L(\mathbf{x})$ and $\epsilon_R(\mathbf{x})$ grow (Fig. 6·2) and can be thought of as occlusion indicators in I_L and I_R, respectively. The above energy terms, by introducing a penalty at each occlusion point, keep the total area of occlusions from growing indefinitely. Otherwise, all image points declared as occluded would result in a low-energy, but degenerate, solution. Since minimization of these terms encourages $D(z)$ close to 1, the computed vector fields are also forced to be close inverses of each other, and thus local outliers are prevented.

Final cost function: In order to perform a joint disparity estimation, implicit occlusion detection and disparity extrapolation, we combine the above energy terms and carry out the following minimizations:

$$\begin{aligned} &\min_{\mathbf{d}_L} E_L, \quad E_L = E_L^P + \eta E_L^S + \mu E_L^O, \\ &\min_{\mathbf{d}_R} E_R, \quad E_R = E_R^P + \eta E_R^S + \mu E_R^O \end{aligned} \tag{6.12}$$

where η and μ are regularization factors. Note that although minimized independently, energies E_L and E_R are coupled through weights $D(\epsilon_L)$ and $D(\epsilon_R)$ used in (6.9) and (6.11). The functionals are minimized in an interleaved fashion; $\mathbf{d}_L$ is assumed constant when computing $\mathbf{d}_R$ and vice versa. A derivation of Euler-Lagrange equations in this case can be found in the Appendix B.

6.4 Why minimize E_L and E_R separately?

We decided to minimize E_L and E_R simultaneously but independently; we minimize E_L for one iteration and then use the estimated $\mathbf{d}_L$ in the next iteration when minimizing E_R,

then use the estimated $\mathbf{d}_R$ when minimizing E_L and so on.

Instead of separate minimization of E_L and E_R, we also attempted minimization of $E_L + E_R$ with respect to $\mathbf{d}_L$ and $\mathbf{d}_R$, however the results proved inferior. This may seem counter-intuitive, therefore we would like to elaborate on this issue.

A minimization of $E_L + E_R$ may look reasonable from the cost function's point of view. After all, we would like both E_L and E_R to be small (ideally zero), so why not their sum be zero as well?

It turns out that $E_L + E_R$ may lead locally (in occlusion areas) to contradictory constraints. When minimizing $E_L + E_R$, $\mathbf{d}_L(\mathbf{x})$ and $\mathbf{d}_R(\mathbf{x})$ must be computed one after another but this interleaving, unlike in (6.12), takes place at pixel level, or $\mathbf{x}$. Consider minimizing $E_L + E_R$ with respect to $\mathbf{d}_L(\mathbf{x})$. Since E_R^S does not depend on $\mathbf{d}_L$, this minimization is equivalent to the minimization of $E_L^P + \eta E_L^S + \mu E_L^O + E_R^P + \mu E_R^O$. Clearly, compared to the minimization in (6.12) with respect to $\mathbf{d}_L$, there are two additional constraints, *via* energies E_R^P and E_R^O. Suppose that a pixel at $\mathbf{x}$ in I_L is visible in I_R, but the same pixel in I_R is a newly-exposed pixel (has no correspondence in I_L). Then, although $\mathbf{d}_L(\mathbf{x})$ can be accurately found by minimization (6.12), it will be biased when minimizing $E_L + E_R$ because of the $E_R^P + \mu E_R^O$ term. In particular, since $\rho_{RL}(\mathbf{x})$ (6.7) is fixed (depends on $\mathbf{d}_R$ only), we have $E_R^P + \mu E_R^O = \alpha D(\varepsilon_R(\mathbf{x})) + \mu(1 - D(\varepsilon_R(\mathbf{x}))) = (\alpha - \mu)D(\varepsilon_R(\mathbf{x})) + \mu$, where $\alpha = \rho_{RL}(\mathbf{x})$ is large because the pixel at $\mathbf{x}$ in I_R is newly exposed. Assuming that $\alpha > \mu$, this constrains $D(\varepsilon_R(\mathbf{x}))$ to be small or, equivalently, $\varepsilon_R(\mathbf{x})$ to be large, causing a forward/backward vector mismatch. This is in contradiction to the constraints imposed by $E_L^P + \mu E_L^O$, and leads to erroneous results in occlusion areas.

However, if both pixels (in I_L and I_R) are visible, no bias takes place when minimizing $E_L + E_R$ since $\alpha = \rho_{RL}(\mathbf{x}) \approx 0$ and thus $E_R^P + \mu E_R^O$ is minimized by $D(\varepsilon_R(\mathbf{x})) \approx 1$ or, equivalently, by $\varepsilon_R(\mathbf{x}) \approx 0$ (no forward/backward vector mismatch). This is consistent with the constraint imposed by $E_L^P + \mu E_L^O$. Clearly, erroneous solutions occur only around occlusions, which we confirmed experimentally.

We can also notice contradictory constraints from the total cost function's point of view.

Consider this: Let us assume that $u_L(\mathbf{x})$ achieved a value that is close to the true value for this position, i.e., $\rho_{LR}(\mathbf{x}) = 0$, $\epsilon_L(\mathbf{x}) = 0$, $F_{\mathbf{x}}(u_L, I_L) = 0$ and, finally, since $\epsilon_L(\mathbf{x}) = 0$, 1-$D_L(\epsilon_L(\mathbf{x})) = 0$. Therefore, all penalty terms originating from I_L are minimized.

Now, consider that the corresponding $\mathbf{x}$ in the right image is an occlusion point i.e., $\epsilon_R(\mathbf{x}) > 0$. Even if this $u_R(\mathbf{x})$ value somehow minimizes $\rho_{RL}(\mathbf{x})$ and $F_{\mathbf{x}}(u_R, I_R)$, since $\epsilon_R(\mathbf{x}) > 0$, there will definitely be some penalty due to the $1 - D(\epsilon_R(\mathbf{x}))$ term.

If all these prediction, smoothness and occlusion penalty terms are summed together, the total cost would not be zero (because $\epsilon_R(\mathbf{x}) > 0$). Therefore, both $u_L(\mathbf{x})$ and $u_R(\mathbf{x})$ should evolve so that the total cost is further minimized. Then, obviously, $u_L(\mathbf{x})$ will deviate from the close-to-true value it had reached. However, if we separate the cost functions, then $u_L(\mathbf{x})$ would stay at the optimal solution and would not be biased toward a wrong solution.

Our experimental results also confirmed that minimization of $E_L + E_R$ leads to inferior results.

6.5 Implementation and experimental results

We discretized the resulting partial differential evolution equations using finite differences (see (Perona and Malik, 1990) for the discretization of anisotropic diffusion). We used an explicit discretization scheme for its simplicity, and a small time step ($\Delta t = 1.5 \times 10^{-5}$) to assure stability of calculations. All subpixel (non-integer position) values, e.g., $I_R(\mathbf{x} + \mathbf{d}_L(\mathbf{x}))$, were computed using bicubic interpolation.

We used a hierarchical implementation to avoid local minima. Images were prefiltered with a Gaussian filter and downsampled so that at the lowest resolution the maximum disparity did not exceed 1-2 pixels. The estimation was started at the lowest resolution and the result propagated to the next higher resolution by interpolation. We used rectified stereo pairs, i.e., $\mathbf{d} = [u\, 0]^T$; the vertical disparity component was set to zero.

In order to carry out evaluation of the proposed algorithm, we introduce two different weighting functions D in our energy formulation: $D_1(z) = 1/(1 + K_1 z^2)$ which weights

Table 6.1: Optical flow (OF) estimation algorithms tested.

Algorithm	K_1	K_2	$g(z)$
Original OF (Horn and Schunck, 1981)	0	0	1
Edge-preserving OF (Alvarez et al., 2002b)	0	0	monotonically-decreasing
Symmetric OF (Alvarez et al., 2002a)	0	>0	monotonically-decreasing
Proposed algorithm	>0	>0	monotonically-decreasing

the photometric error in (6.9) and $D_2(z) = 1/(1 + K_2 z^2)$ which keeps the total area of occlusions from growing indefinitely (6.11). As shown in Table 6.1, for different values of K_1 and K_2, and different functional forms of $g(z)$ our formulation may be simplified to the original optical flow (Horn and Schunck, 1981), edge-preserving optical flow (Alvarez et al., 2002b), or symmetric optical flow (Alvarez et al., 2002a) estimation, the latter one forcing the two disparity fields to be close inverses of each other. The symmetric optical flow algorithm includes energy E_3 (6.11) but does not disable the data-matching term in (6.9). This is of interest for state-of-the-art video coding based on the discrete wavelet transform (DWT) as it is able to ensure a close invertibility of vector fields, important for such coders (Konrad and Božinović, 2005). Also, note that for K_1=K_2=0 and $g(z)$=1, minimizations in (6.12) reduce to two original optical flow algorithms executed in parallel. In all experiments, whenever K_1 and K_2 are non-zero we use the value of 10, while η=6000, μ=2000.

First, we tested the four optical flow approaches on two synthetic sequences. Fig. 6·3 shows an unusually-shaped object which is displaced horizontally by 15 pixels over a stationary background. The original images and corresponding ground-truth occlusion area for I_L are in the top row of Fig. 6·3. The ground-truth disparity map for I_L and its four estimates, presented as intensity, as well as the recovered occlusions, are shown in the remaining two rows.

The second synthetic sequence (Fig. 6·4) is more challenging; two circles displace in opposite directions. There are three occlusion regions between images and a significant

Table 6.2: Absolute error per pixel in computed disparity fields.

	Image #1 (Fig. 6·3)		Image #2 (Fig. 6·4)	
	u_L	u_R	u_L	u_R
Original OF	4.57	4.67	1.63	1.44
Edge-preserving OF	1.55	1.51	0.81	0.52
Symmetric OF	1.61	1.83	0.60	0.45
Proposed algorithm	0.58	0.53	0.35	0.36

portion of occlusions is due to one object covering the other. Table 6.2 shows the absolute error per pixel for the estimated disparities. The error is estimated as follows: $\frac{1}{NM}\sum_{x=1}^{N}\sum_{y=1}^{M}||\mathbf{d}_{est}(x,y)-\mathbf{d}_{true}(x,y)||$, where N and M are numbers of columns and rows respectively.

It is clear that the proposed method (Figs. 6·3.h and 6·4.h) outperforms the original (Figs. 6·3.e and 6·4.e) and edge-preserving optical flow (Figs. 6·3.f and 6·4.f) algorithms, both subjectively and numerically. Note a significant improvement offered by the edge-preserving regularization compared to the original optical flow algorithm. The symmetric optical flow algorithm (Figs. 6·3.g and 6·4.g) offers some subjective and numerical advantage over the edge-preserving optical flow but since it enforces forward/backward vector consistency at occluded pixels, the improvement is limited. Had the occlusion areas been very small, the symmetric optical flow would have improved the results significantly (Alvarez et al., 2002a). In our images, however, disparity mismatch over large occlusion areas affects visible pixels through diffusion and results in disparity errors. Still, this method is of interest for DWT-based video coding due to close mutually-inverse properties of the resulting vector fields (Konrad and Božinović, 2005).

Table 6.3: The threshold *vs.* symmetric difference of true occlusion areas and estimated occlusion areas.

ζ	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
Symmetric difference	1515	1577	1615	1646	1666	1688	1725	1782	1886

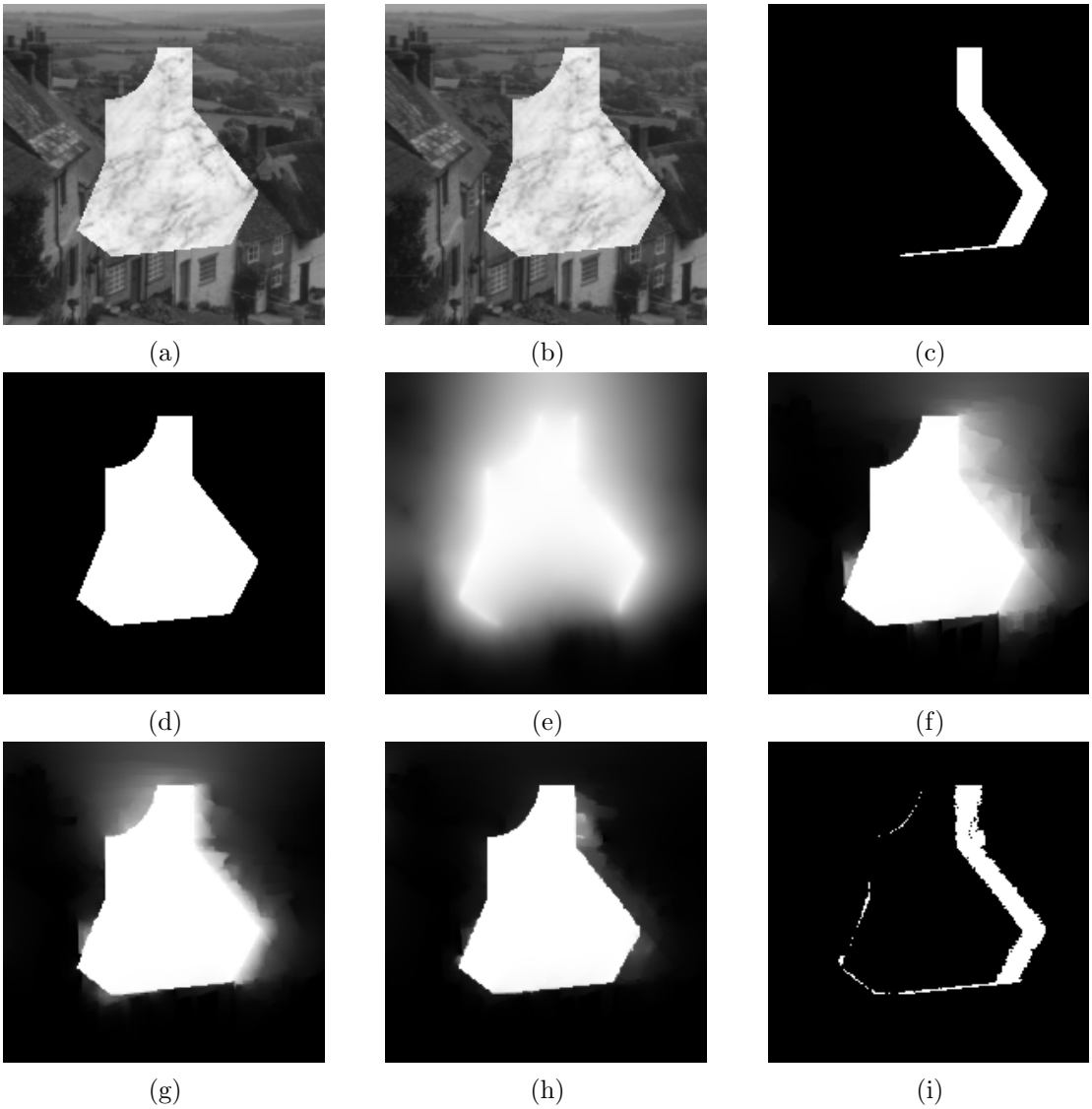


Figure 6·3: Results for a computer-generated pair of images: (a) I_L; (b) I_R; ground-truth: (c) occlusions for I_L and (d) disparity for I_L; and disparities for I_L computed using progressively more complex formulations: (e) original OF; (f) edge-preserving OF; (g) symmetric OF; (h) proposed method; and (i) likely occlusion areas obtained by thresholding $1-D(\epsilon_L(\mathbf{x}))$. In disparity images, black and white graylevels represent 0 and 15 pixels of disparity, respectively.

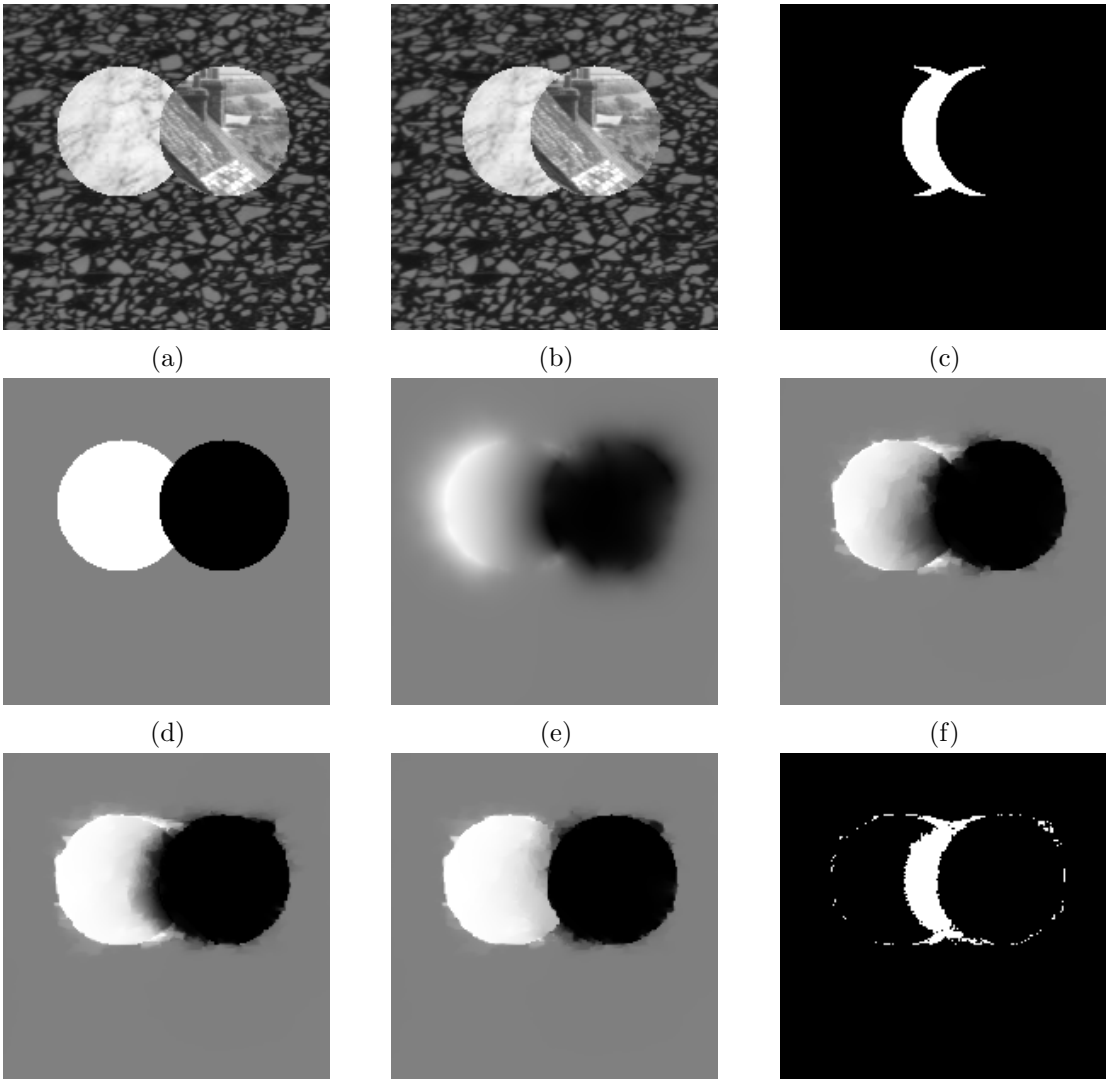


Figure 6·4: Results for a computer-generated pair of images: (a) I_L; (b) I_R; ground-truth: (c) occlusions for I_L and (d) disparity for I_L; and disparities for I_L computed using progressively more complex formulations: (e) original OF; (f) edge-preserving OF; (g) symmetric OF; (h) proposed method; and (i) likely occlusion areas obtained by thresholding $1 - D(\epsilon_L(\mathbf{x}))$. In the disparity images, black, gray and white graylevels represent -10, 0, and 10 pixels of disparity, respectively.

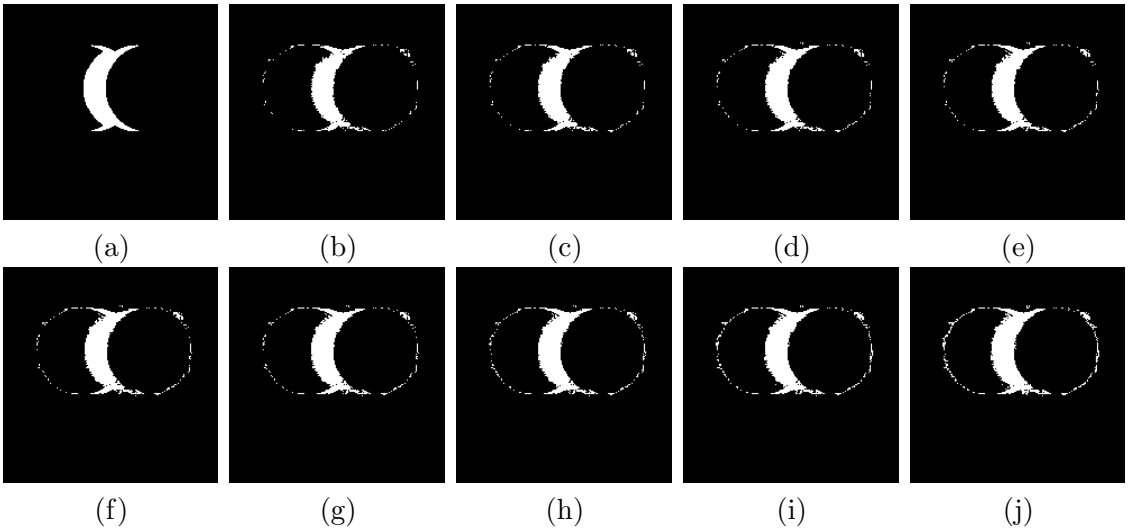


Figure 6·5: (a) True occlusion areas; occlusion areas estimated at different threshold values (b) $\zeta = 0.9$ (c) $\zeta = 0.8$ (d) $\zeta = 0.7$ (e) $\zeta = 0.6$ (f) $\zeta = 0.5$ (g) $\zeta = 0.4$ (h) $\zeta = 0.3$ (i) $\zeta = 0.2$ (j) $\zeta = 0.1$. Symmetric difference between true and estimated disparity ranges from 1515 at $\zeta = 0.9$ to 1886 at $\zeta = 0.1$ (Table 6.3).

Figures. 6·3.i and 6·4.i show thresholded values of $1 - D(\epsilon_L(\mathbf{x}))$. White areas show $1 - D(\epsilon_L(\mathbf{x})) > \zeta$, which are the likely occlusion areas. In our experiments, we use $\zeta = 0.9$, thus we assume that if $D(\epsilon_L(\mathbf{x})) < 0.1$, then $\mathbf{x}$ is likely an occlusion area. The reason for this specific value of 0.1 is simple: at $K = 10$, if $D(\epsilon_L(\mathbf{x})) < 0.1$, then $||\epsilon_L(\mathbf{x})|| >$ 1 pixel, which is enough to warrant an occurrence of an occlusion. Although ζ seems to be another parameter to set, we would like to show that the results are not sensitive to this parameter. Figure 6·5 show the estimated occlusions for various values of ζ and Table 6.3 show symmetric difference of true occlusion areas and estimated occlusion areas. It is clear that results in Fig. 6·5 are visually indistinguishable and symmetric difference values are similar. The reason is that in occlusion areas, $1 - D(\epsilon_L(\mathbf{x}))$ is very close to one (because $D(\epsilon_L(\mathbf{x}) \approx 0)$ whereas in visible areas it is very close zero, therefore any ζ between zero and one results in similar estimates.

We also compared the four optical flow approaches in the presence of noise; we added zero-mean white Gaussian noise to the test image from Fig. 6·3. The absolute disparity

Table 6.4: Absolute disparity error per pixel for u_L on test image from Fig. 6·4 at different levels of zero-mean white Gaussian noise.

Resulting PSNR(dB)	Original OF	Edge-preserving OF	Symmetric OF	Proposed algorithm
No noise	1.63	0.81	0.60	0.35
27.01	1.66	0.91	0.74	0.50
24.09	1.69	1.08	0.81	0.60
23.12	1.69	1.00	0.87	0.64
20.35	1.80	1.18	0.96	0.72

error per pixel for u_L is shown in Table 6.4 for different levels of noise. Clearly, the proposed method performs well under noise as well. This can be explained by the adaptive nature of the algorithm; since disparities at noisy pixels usually lead to significant geometric errors (6.8), the contribution from these pixels is disabled in (6.9). It should be also noted that the hierarchical scheme used, which includes a prefiltering step, acts as a noise suppressor and helps all tested methods deal with noise.

Finally, we tested the algorithms on camera-acquired images: *Exit* (Fig. 6·6) and *Michel* (Fig. 6·7). For *Exit*, the improvements are clear in occlusion area to the right of the person closest to the camera, visible especially in close-up images (Fig. 6·6.g-i). Original optical flow result is very unsatisfactory. Although results of the symmetric and edge-preserving optical flow are much better, there is a clear spillover of disparities from person's body into the background in both results (pixels in the occlusion area fail to find correspondence in the other image). However, such errors are largely corrected by the proposed method, because the matching term (6.9) is disabled in occlusion areas. The estimated occlusion areas are shown in Fig. 6·6.j. Similar improvements can be observed for *Michel* in Fig. 6·7. Note the large occlusion areas, e.g., behind the head, lead to incorrect large disparities for symmetric and edge-preserving optical flow (Fig. 6·7.d and 6·7.e), but are corrected by the proposed method (Fig. 6·7.f. Close-ups in Fig. 6·7.g-i clearly show the improvements. In this example, we used $\mu = 5000$, because the occlusion area is much larger; unless the

occlusion count is penalized, it will grow beyond reasonable limits.

6.6 Convergence of energy minimization

In this section, we would like to comment on the convergence of energy minimizations in (6.12). An analytic proof of convergence of the the algorithms minimizing energies E_L and E_R in (6.12) is very difficult and beyond the scope of this thesis. In practice, these algorithms have converged in all our experiments (we used a large but fixed number of iterations). Fig. 6·8 shows the evolution of energies E_L and E_R per pixel against iteration number. These plots show the change in both energies at the final level of hierarchy (i.e., no downsampling) for the synthetic sequence from Fig. 6·4.

It is clear that the energies decrease rapidly in the first few hundred iterations and then reach a relatively steady state. Figs. 6·8.c and d show the last 2000 iterations, where we note that slight oscillations take place. However, the amplitude of oscillations is negligible when compared to the initial energies (around 0.25% and 0.13% of initial energies for E_L and E_R, respectively). To prove that oscillations have negligible effect on the results, we measured the lowest and highest values of E_R in the last 2000 iterations. The lowest value is achieved at iteration #11556, while the maximum value is achieved at iteration #10839. The resulting disparity fields at iterations #10839, #11556 and #12000 (final iteration) are shown in Fig. 6·9. It is clear that the resulting disparities are visually indistinguishable. This suggests that the algorithms can be stopped after a few thousand iterations and that a precise selection of the number of iterations is not critical.

6.7 Parameter selection

As in many other methods, an important issue is one of parameter selection. In the proposed formulation, three parameters K, η and μ influence the results. We chose $\eta = 6000$ and $\mu = 2000$ experimentally. While a larger η would force an even smoother disparity field, a larger μ would further reduce the number of (implicitly) estimated occlusion pixels. We chose K to be 10 because when ϵ, i.e., mismatch between vector fields, is larger than 1

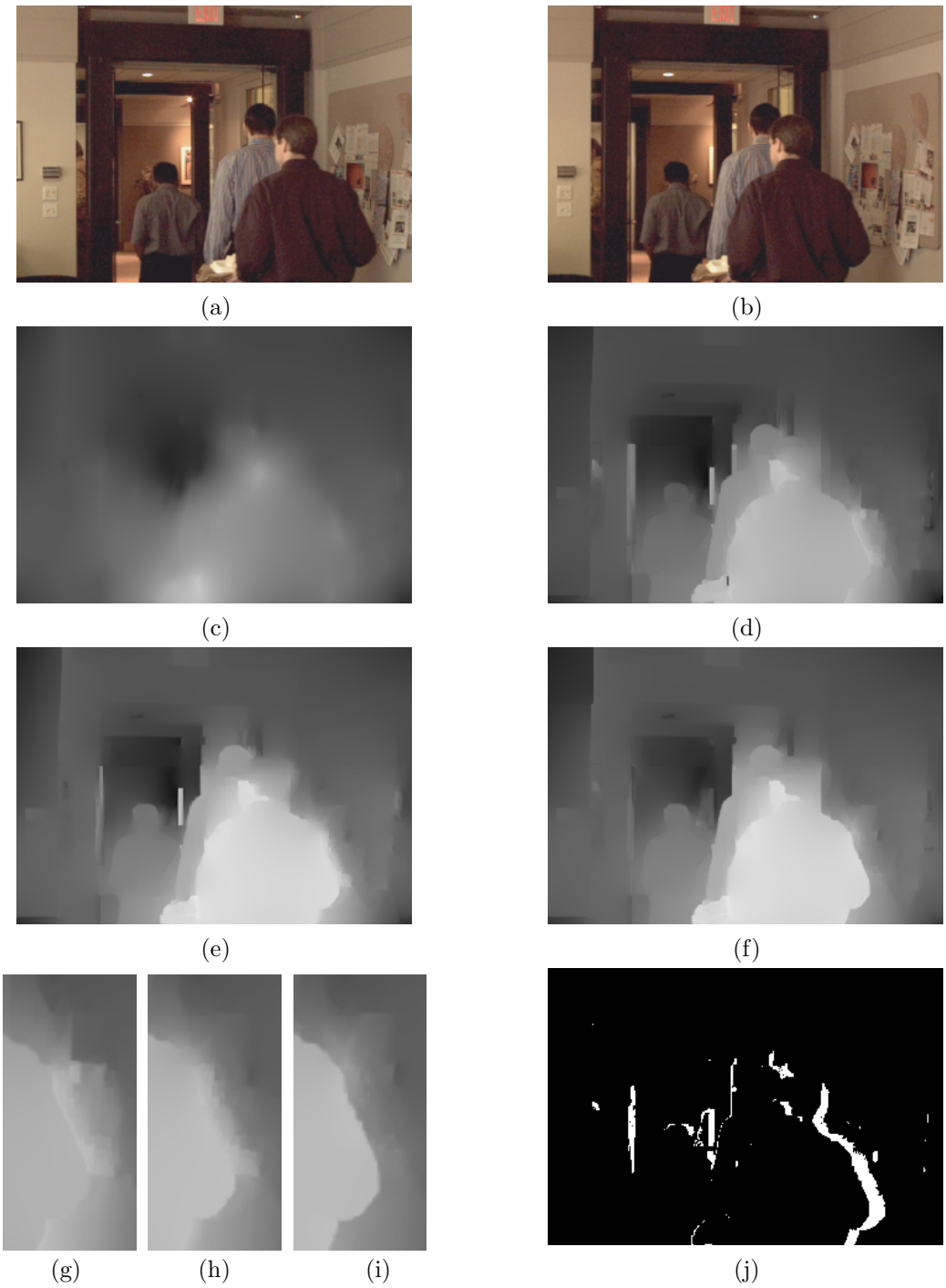


Figure 6·6: Experimental results for *Exit* image pair (property of Mitsubishi Electric Research Labs) : (a) I_L; (b) I_R; estimated disparity for I_R: (c) original-OF; (d) edge-preserving-OF (e) symmetric-OF; and (f) proposed method; and (g-i) close-ups of results from (d-f), (j) likely occlusion areas obtained by thresholding $1 - D(\epsilon_R(\mathbf{x}))$.

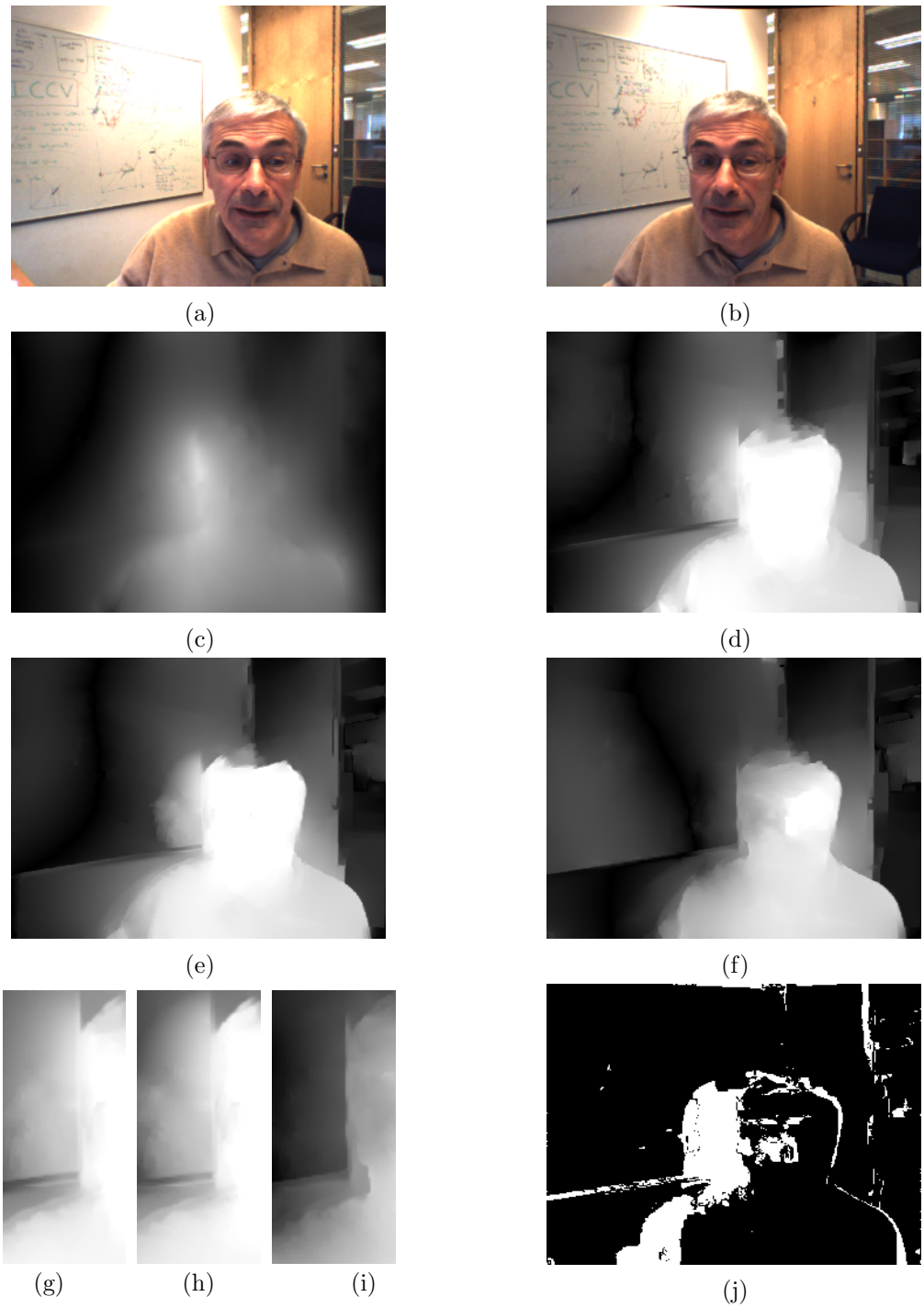


Figure 6·7: Experimental results for *Michel* image pair (property of Microsoft Research Cambridge, UK):(a) I_L; (b) I_R; estimated disparity for I_R: (c) original-OF; (d) edge-preserving-OF (e) symmetric-OF; and (f) proposed method; and (g-i) close-ups of results from (d-f), (j) likely occlusion areas obtained by thresholding $1 - D(\epsilon_L(\mathbf{x}))$.

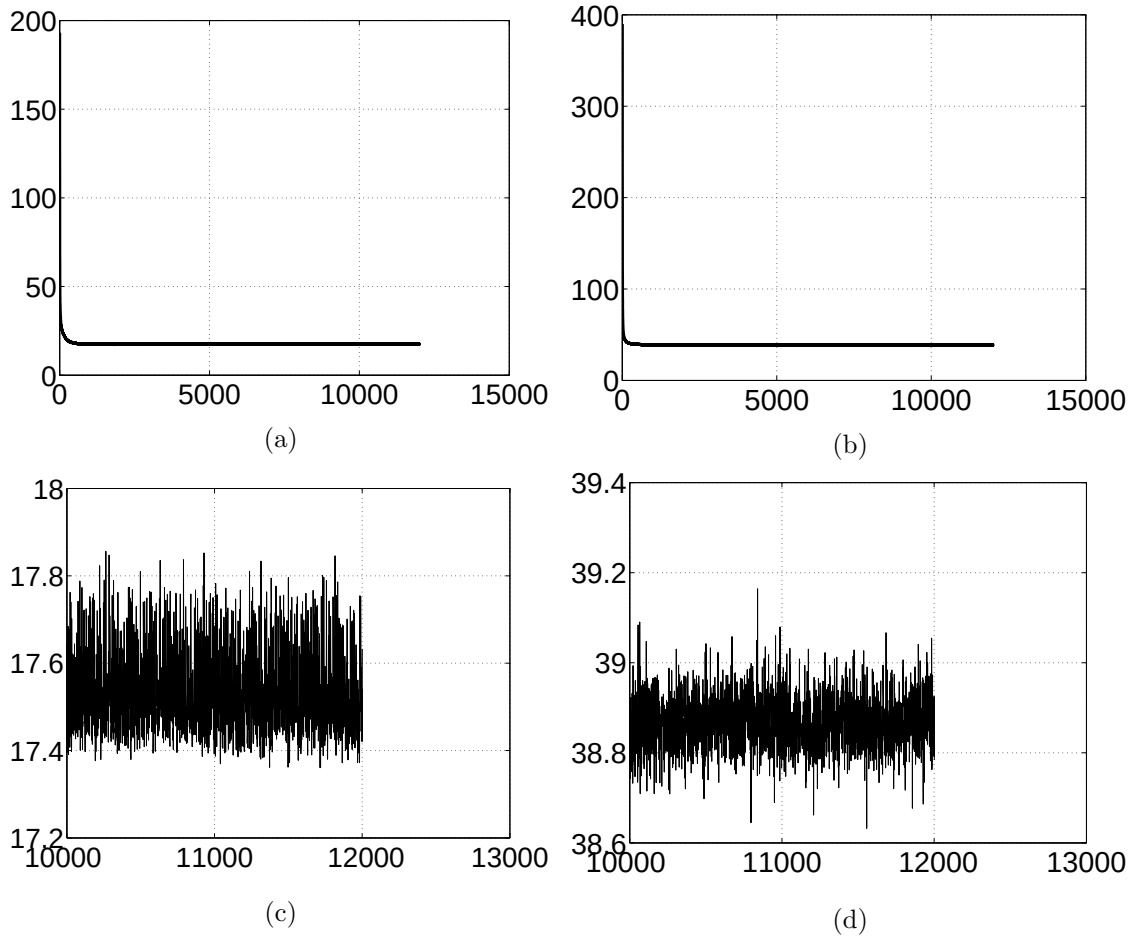
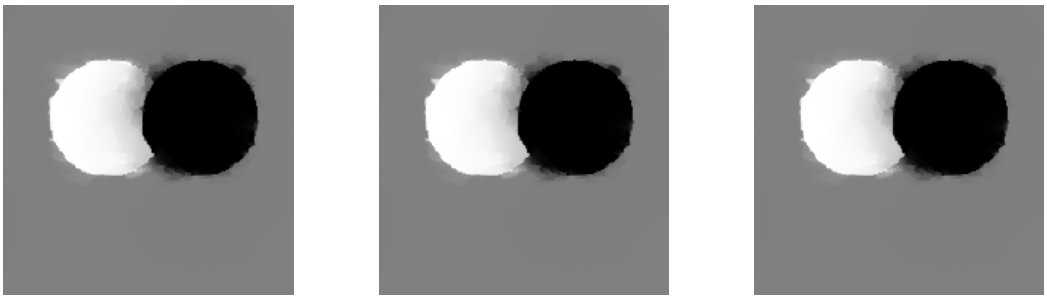


Figure 6·8: Plots of energy per pixel with respect to iteration number for (a) E_L; (b) E_R. Final 2000 iterations are shown for (c) E_L; (d) E_R.

pixel, $D(\epsilon)$ falls below 0.1, a small enough value to significantly reduce contribution of the intensity matching term. Note that there exist methods such as expectation maximization (Dempster et al., 1977), min-max principle (Gennert and Yuille, 1988) and unbiased risk estimator (Ng and Solo, 1997) that can be used to automatically select parameter values.

In order to demonstrate that a very precise selection of parameters is not necessary in our method, Table 6.5 shows the absolute disparity error per pixel for the test image from Fig. 6·4 while changing either K, or η, or μ. It can be seen that as parameters are increased threefold, the error changes at most by 10-20%. We also tested a relatively very small value of $\mu = 100$; almost half of the pixels were marked as occlusion. This was to be



(a) (b) (c)

Figure 6·9: Resulting disparity fields after (a) 10839; (b) 11556 and (c) 12000 iterations.

Table 6.5: Absolute disparity error per pixel for the test image from Fig. 6·4 and different parameter values. In each experiment one parameter is adjusted while other parameters are unchanged.

$\eta = 6000$, $\mu = 2000$

K	u_L	u_R
3	0.52	0.46
7	0.47	0.43
10	0.35	0.36
12	0.37	0.36

$K = 10$, $\mu = 2000$

η	u_L	u_R
1000	0.54	0.45
3000	0.43	0.40
6000	0.35	0.36
9000	0.37	0.37

$K = 10$, $\eta = 6000$

μ	u_L	u_R
100	1.00	1.16
1000	0.53	0.47
2000	0.35	0.36
3000	0.44	0.43

expected since, as we mentioned earlier, this leads to mismatched disparities $\mathbf{d}_L$ and $\mathbf{d}_R$ and, consequently, to disabling of the photometric error (6.9).

6.8 Computational load and limitations of the proposed method

We observed that the proposed approach brings only about 40% of additional computational load to the standard optical flow algorithm due to the additional interpolation operations stemming from terms such as $\widetilde{v}_{\cdot}^{x}$. If the occlusion areas are significant, the number of iterations must be increased so that diffusion can fill-in these areas using neighboring values. One shortcoming of the method is evident in highly-textured images because image-driven disparity diffusion is inhibited due to the high local intensity gradient; a common problem of image-driven regularizers.

6.9 Conclusions

In this chapter, we presented a variational framework for joint disparity estimation, occlusion detection, and disparity extrapolation based on two images only. The new formulation calculates two closely-symmetric disparity fields and also inpaints the disparity in occlusion areas. The proposed algorithm shows significant improvement over original and edge-preserving optical flow formulations both subjectively and numerically. Moreover, the symmetric variant of the proposed algorithm may be interesting for DWT-based video coding because of the particular relationship between the resulting vector fields (close mutual inverses).

By the end of this chapter we proposed methods to solve the challenges that we mentioned at the beginning of this dissertation. In the next chapter, we will utilize these tools to reconstruct intermediate views.

Chapter 7

Occlusion-aware spline-based view reconstruction

We discussed the main challenges in view reconstruction in Chapter 3. These were: estimation of disparity and occlusions, and formation of intermediate view. In Chapters 4, 5 and 6, we proposed various methods to address each of these challenges. In this chapter, we combine the proposed methods in a novel occlusion-aware intermediate view reconstruction algorithm. The chapter will start with an introduction to occlusion awareness in view reconstruction, and then will discuss how the proposed methods can be used to achieve this goal. Finally, we will show experimental results on both synthetic and real images.

7.1 Introduction

Let us first review the steps required in occlusion-aware view reconstruction from two images captured in a small-baseline stereo setup.

Intermediate view reconstruction can essentially be separated into five steps, hence five challenges to solve, as follows:

1. Find the structure: Estimation of disparity field or depth map
2. Find the innovation areas or problematic areas: Estimation of occlusion areas
3. Handle the problematic areas: Recover (extrapolate) disparity in occlusion areas
4. Estimate visibility of points in intermediate view
5. Estimate texture of intermediate view.

Although it is possible to handle each of these steps separately, one of the ideas in our work was to combine as many sub-components as possible into the same formulation. The

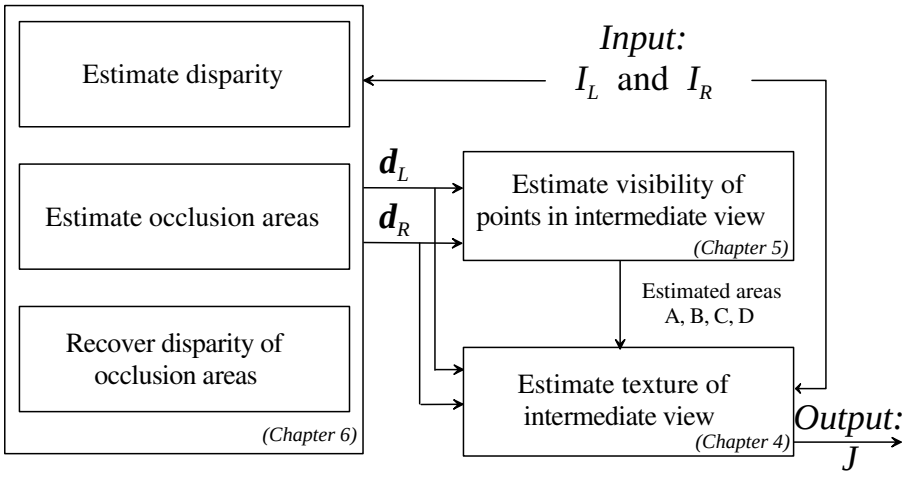


Figure 7·1: Block diagram showing the proposed method as well as input and output of each step.

reason is that we have observed that the above sequence of steps usually leads to sub-par reconstructions.

One source of sub-par results is that each step introduces errors which are propagated or even magnified in subsequent steps. Also, the unknown disparity and partitioning into occluded/newly-exposed/visible areas are closely related. For example, disparities can be used to infer occlusion information, while occlusion information is required to estimate reliable disparities. Therefore, interaction between these unknowns is essential during their estimation.

7.2 Proposed reconstruction method

A block diagram explaining the proposed method is shown in Fig. 7·1. The five steps described earlier are implemented as follows:

Steps #1, 2, 3: In view of a need for joint estimation, we use the occlusion-aware disparity estimation algorithm proposed in Chapter 6 in order to accomplish the first three steps above.

Although the occlusion-aware disparity estimation recovers disparities and occlusion areas, they are both defined on lattice Λ of either I_L or I_R, whereas in order to reconstruct

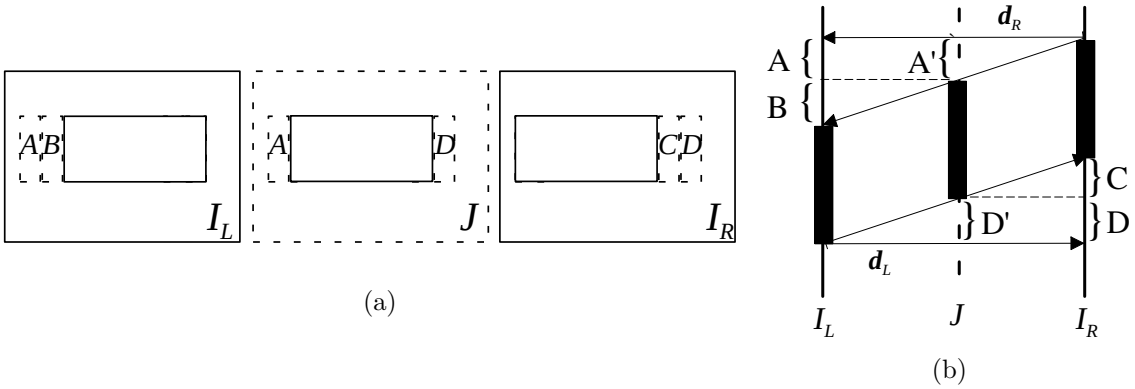


Figure 7·2: Illustration of the need of selective forward-mapping. (a) Three images and (b) their cross sections. Areas A and D are visible intermediate view despite being occluded between I_L and I_R.

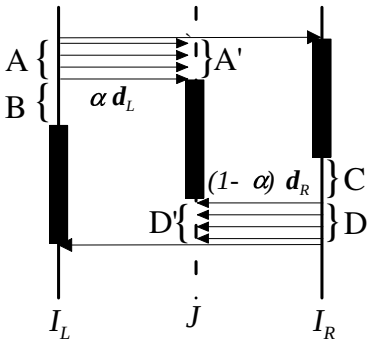
an intermediate view, visibility and occlusion labels of the points are needed on lattice Λ in the intermediate view J.

Step #4: The estimation of visibility of points is a crucial step. We briefly discussed this problem in Section 5.4. We would like to illustrate this problem with an example. Consider images in Fig. 7·2.a and their cross-section in Fig. 7·2.b where an object displaces between I_L and I_R. For the sake of simplicity let us assume that the background has zero disparity, i.e., does not move between I_L and I_R.

The occlusion-aware optical flow can estimate reliable disparities for both images as well as occlusion areas $\{A \cup B\}$ in I_L and $\{C \cup D\}$ in I_R. However, the important fact is that the occlusion area $\{A \cup B\}$ from I_L is *not* fully occluded in J. As it can be noticed, A' (whose texture can only be derived from A in I_L) is visible in J. Similarly, the occlusion area $\{C \cup D\}$ from I_R is not fully occluded in J. The area D' (whose texture can only be derived from D in I_R) is visible in J.

However, the disparity estimation stage cannot distinguish these partially visible areas. It will only estimate $\{A \cup B\}$ and $\{C \cup D\}$. Therefore we need a mechanism to estimate the visibility of points, or in other words, to identify each area.

In order to compute the pixel visibility in J, we will use the method proposed in Section 5.3. One of the most important properties of this algorithm is that it allows to estimate



(c)

Figure 7·3: Selective forward compensation. All points of I_L and I_R are forward disparity-compensated except areas B and C.

the areas that *will* be occluded/exposed off the domain of source images I_L and I_R at an intermediate position $0 < \alpha < 1$ (0 being I_L and 1 being I_R, respectively). Therefore, this method permits identification of pixels that will be occluded in the intermediate view. For example, in Fig. 7·2.b, the algorithm can detect areas D' and $\{C \cup D\}$ by using vector fields $\alpha\mathbf{d}_L$ and $\mathbf{d}_L$ respectively. This topic was previously covered in Section 5.4.

We want all points from I_L and I_R to be used in texture estimation in J except the areas B and C, because they are occluded in the intermediate view. Therefore, our final task in this step is to detect and eliminate these areas. This can be achieved through the following relationship (Fig. 7·3):

$$\forall \mathbf{x} \in \{A \cup B\} \;:\; \begin{cases} \mathbf{x} \in A & \text{if } (\mathbf{x} + \alpha\mathbf{d}_L) \in A' \\ \mathbf{x} \in B & \text{otherwise} \end{cases} . \tag{7.1}$$

A similar relationship can be written for C and D in I_R as follows:

$$\forall \mathbf{x} \in \{C \cup D\} \;:\; \begin{cases} \mathbf{x} \in D & \text{if } (\mathbf{x} + (1-\alpha)\mathbf{d}_R) \in D' \\ \mathbf{x} \in C & \text{otherwise} \end{cases} . \tag{7.2}$$

These relationships test whether a forward-compensated pixel from I_L (7.1) or I_R (7.2) belongs to visible or occluded area. In other words, we would like to fill in the texture of exposed area by using the texture from the other input image. For example, since area A'

is exposed from I_R to J, this area must be predicted from A of I_L. On the other hand, B of I_L will be occluded on J and thus must be excluded from the reconstruction. The relations (7.1-7.2) allow us to identify which point will be visible or be occluded. Note that for the special case of static background (i.e., $\mathbf{d}_L = \mathbf{d}_R = 0$ in A and D), A and D are equal A' and D', respectively.

One may ask why disparity of $A \cup B$ would be reliable. They are expected to be reliable with the help of the occlusion-aware optical flow method. These areas will be assigned plausible disparities *via* diffusion.

This step is the crucial step in the occlusion awareness. In contrast, pivoting-based reconstruction, for example, is hampered by occlusions as illustrated in Fig. 7·4. A typical field estimated using pivoting in J and spatial regularization results in a "rubber" effect (no sharp boundary between the object and the background). The reason is that the absence of an underlying image makes edge-preserving regularization impossible. Additionally, it is not clear how to estimate occlusion by using a single disparity field. Even if occlusions were somehow estimated, assigning disparity values to these areas would be impossible as well, again due to absence of underlying image (no diffusion is possible). Finally, estimating the visibility of points would not be possible, because given a single disparity field, there is not enough information to infer this data.

Steps #5: The last step is forward disparity compensation and, consequently, spline-based reconstruction as proposed in Chapter 4. However, as we have mentioned, the areas of I_L and I_R that are going to be occluded in the intermediate image must be excluded from the reconstruction. Thanks to the previous step, areas to be occluded in J have been identified and we are free to choose areas in I_L and I_R to forward-compensate onto J as illustrated in Fig. 7·3.

Therefore, we propose to *selectively* forward-compensate only what will be visible in the intermediate view (i.e., A of I_L and D of I_R), and eliminate occlusion areas (i.e., B of I_L and C of I_R). In order to recover J from the resulting irregular intensities, we slightly

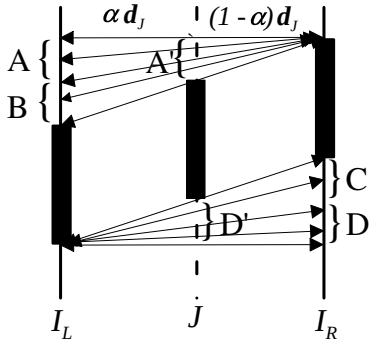


Figure 7·4: Illustration of occlusion effects on a horizontal cross-section (single row) of I_L, J, I_R in pivoting-based reconstruction. Typical (incorrect) disparity fields $\mathbf{d}_L$ and $\mathbf{d}_R$ computed under spatial regularization in presence of occlusions.

modify the formulation in (4.9-4.11) such that occlusion areas are excluded as follows:

$$P_n^L = \{\mathbf{x}_n + \alpha \mathbf{d}_L(\mathbf{x}_n), I_L(\mathbf{x}_n)\} \text{ for } n = 1, ..., N_L, \text{ and } \mathbf{x}_n \in \Lambda_L \backslash B \tag{7.3}$$

$$P_m^R = \{\mathbf{x}_m + (1-\alpha) \mathbf{d}_R(\mathbf{x}_m), I_R(\mathbf{x}_m)\} \text{ for } m = 1, ..., N_R, \text{ and } \mathbf{x}_m \in \Lambda_R \backslash C \tag{7.4}$$

$$P = P^L \cup P^R \tag{7.5}$$

The difference is that the domains are changed from Λ_L to $\Lambda_L \backslash B$ and Λ_R to $\Lambda_R \backslash C$ where B and C are occlusion areas of each image. The idea behind this selective projection is that unless occlusion areas are eliminated, intermediate views will exhibit "double-texture" effects, because occluded and visible areas will overlap in the intermediate view.

The overconstrained spline-based method proposed in Chapter 4 will estimate an optimal function $f(\mathbf{x})$ that minimizes cost function E (Section 4.2.2) for the set P in 7.5.

7.3 Experimental results

In this section we compare our results for occlusion-aware and unaware view reconstruction. Specifically, we compare the following:

1. Pivoting-based reconstruction: Method introduced in Chapter 4, which is occlusion-unaware

2. Spline-based reconstruction (Fig. 7·1) that uses occlusion-aware optical flow estimation and handles visibility of points; this method employs all steps listed in Section 7.2.

In order to show benefits of the proposed approach, we constructed a synthetic image set with significant occlusions, shown in Figs. 7·5.a-c. The object displaces 19 pixels between I_L and I_R on a static background. The ground-truth disparities for left and right images are shown in Figs. 7·5.d and e. The ground-truth occlusion areas between the left and right images are shown in Figs. 7·5.f and g.

The computed disparity fields using occlusion-aware optical flow algorithm proposed in Chapter 6 are shown in Figs. 7·5.h and i. Note the sharp discontinuities in the disparity field and accurately inpainted disparity in occlusion areas. Estimated occlusions are shown in Figs. 7·5.j and k, and are very accurate when compared to the true occlusion areas.

Figures. 7·5.l-m show the areas that will be occluded in the intermediate view computed using the newly-exposed area detection algorithm proposed in Section 5.3 (areas corresponding to B and C in Fig. 7·3). Although there are a few false positives, such errors will have minimal impact on the reconstruction quality due to overconstraining the solution (texture in the intermediate view will be forward-projected from at least one source image). The final reconstruction is shown in Fig. 7·5.n (to be compared with Fig. 7·5.b). The difference between ground-truth and spline-based intermediate view is shown in Fig. 7·5.o. The PSNR value is 31.59dB.

Since image-driven anisotropic diffusion (Nagel and Enkelmann, 1986) is not possible in pivoting-based methods, the resulting disparity field (Fig. 7·5.p) can neither capture the shape of the object nor give any occlusion information. The pivoting-based reconstruction is shown in Figs. 7·5.q (to be compared with Fig. 7·5.b) with a PSNR value of 26.77dB. Numerically, the spline-based reconstruction outperforms the pivoting-based approach by almost 5dB. Although visible points have comparable reconstruction quality in both approaches, occlusion areas show a significant improvement in spline-based reconstruction as is clear from prediction error images in Figs. 7·5.o and r. The texture in the background

is properly reconstructed in the spline-based method but distorted in the pivoting-based reconstruction. The only problem that can be noticed in the spline-based result is on object boundary, where larger errors are present. The reason for this is that occlusion and disparity estimation results, although very good, are not perfect.

Finally, Fig. 7·5.s shows prediction error for spline reconstruction if step #4 is skipped. As it can noticed, there is a significant distortion in occlusion areas. This is due to the double-texture effect. Since the to-be-occluded texture (marked with red in Fig. 7·5.l and m) is not eliminated, the final reconstructed image had conflicting data in the occlusion area (both object and background texture). The reconstructed image has PSNR is 25.83dB.

Reconstruction results with explicit occlusion handling improves results on real-world images as well. The stereo sequence in Fig. 4·4 showed significant artifacts without occlusion handling. Let us show the results when occlusions are properly handled.

Disparity fields estimated for this sequence using occlusion-aware optical flow are shown in Fig. 7·6.c and d. When compared to Fig. 4·4.c, these disparity maps preserve object shapes successfully and are subjectively more accurate. The occlusion areas estimated by using the proposed approach in step #4 are shown in Fig. 4·4.e and f. In Fig. 4·4.f, the occlusions marked on the bottom left of the image (on the table) seem to be incorrect, however, these are due to the shadow of the speaker, which have the same disparity with speaker.

The pivoting-based reconstruction (reproduced from Fig. 4·4.d) and occlusion-aware spline-based reconstruction are shown in Fig. 7·6.g and h, respectively. It is clear that object shapes are well preserved and spline-based reconstruction outperforms pivoting-based reconstruction (shown in closeups Fig. 7·6.i and j).

The closeups of the texture around occlusion area from left image, pivoting-based and spline-based reconstruction also show that spline-based reconstruction preserves texture better than pivoting-based approach. The arm is 'split' in pivoting-based approach (Fig. 7·6.l) but is properly reconstructed in spline-based approach (Fig. 7·6.m). The improvements are due to more accurate disparity fields and successful occlusion handling.

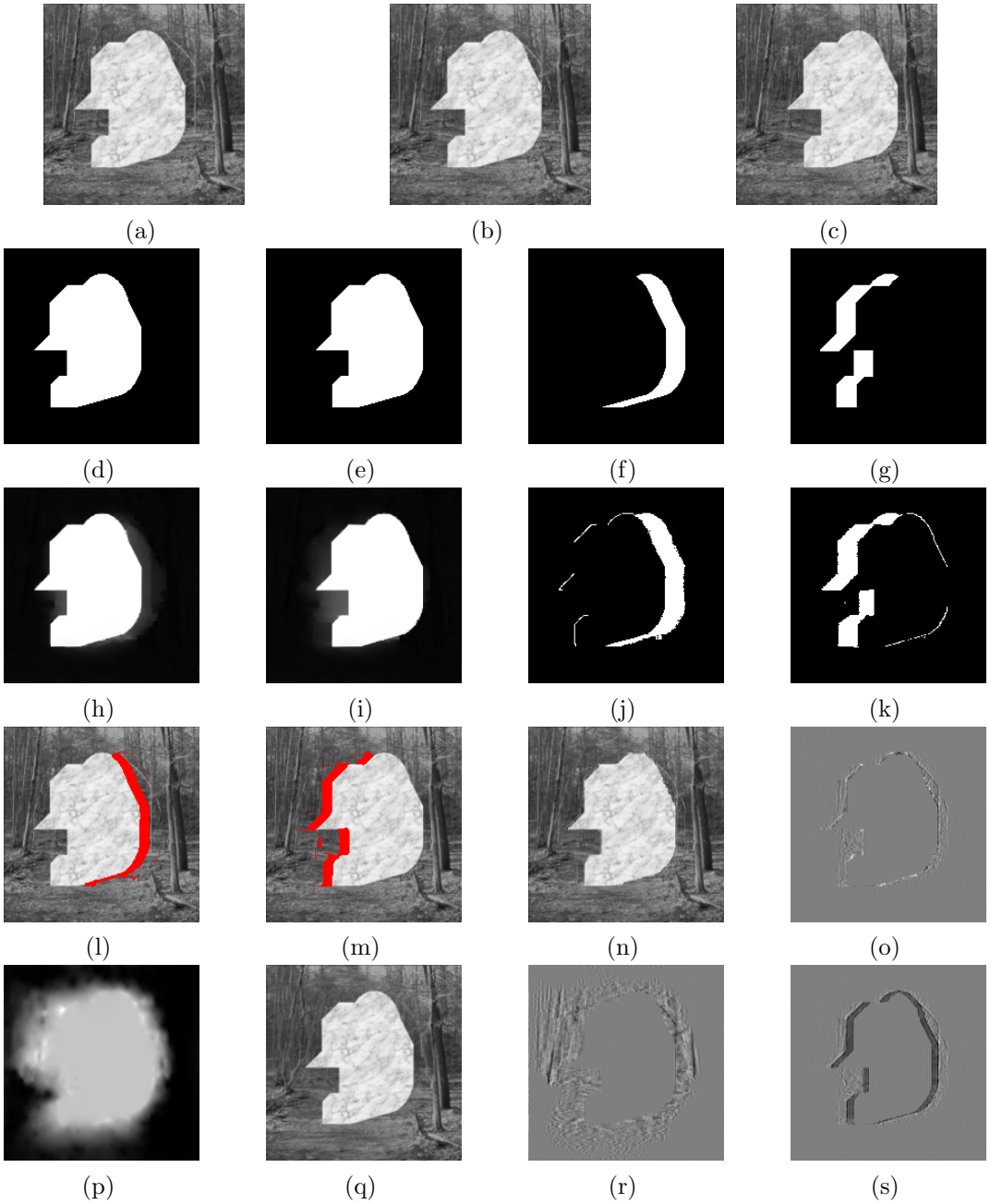


Figure 7·5: Original images: (a) left (I_L), (b) intermediate (J), and (c) right (I_R); true disparity: (d) left-to-right, and (e) right-to-left ; true occlusions: (f) left-to-right, and (g) right-to-left; estimated OF: (h) left-to-right ($\mathbf{d}_L$), and (i) right-to-left ($\mathbf{d}_R$); estimated occlusions: (j) left-to-right, and (k) right-to-left; pixels to be occluded in midpoint image J that come from: (l) left image (I_L), and (m) right image (I_R); (n) spline-based reconstruction (31.59dB), and (o) its error; (p) disparity for the pivoting-based approach; (q) pivoting-based reconstruction (26.77dB), and (r) its error; (s) error of spline-based reconstruction without step #4 (25.83dB).

However, as in the synthetic sequence, there are a few small artifacts on the edges of the objects (for example on left edge of the speaker). This is again due to less-than-perfect disparity and occlusion estimates. Nevertheless, the gain in overall quality makes these small errors insignificant.

Finally, we show the views reconstructed for Ballroom sequence (property of Mitsubishi Electric Research Laboratories). Figures 7·7.a and f show left and right images and Fig. 7·7.b-e show reconstructed intermediate views. Again, the intermediate views are of high quality.

7.4 Conclusions

In this chapter, we proposed an occlusion-aware spline-based intermediate view reconstruction algorithm. In combination with occlusion-aware disparity estimation, it produces much better results visually on data sets with significant occlusions than the pivoting-based approach. The main advantage of the method stems from accurate disparity fields which can utilize the underlying image to account for object boundaries. Since occlusions are properly handled, the spline-based results do not exhibit the 'rubber' effect, typical of pivoting-based methods.

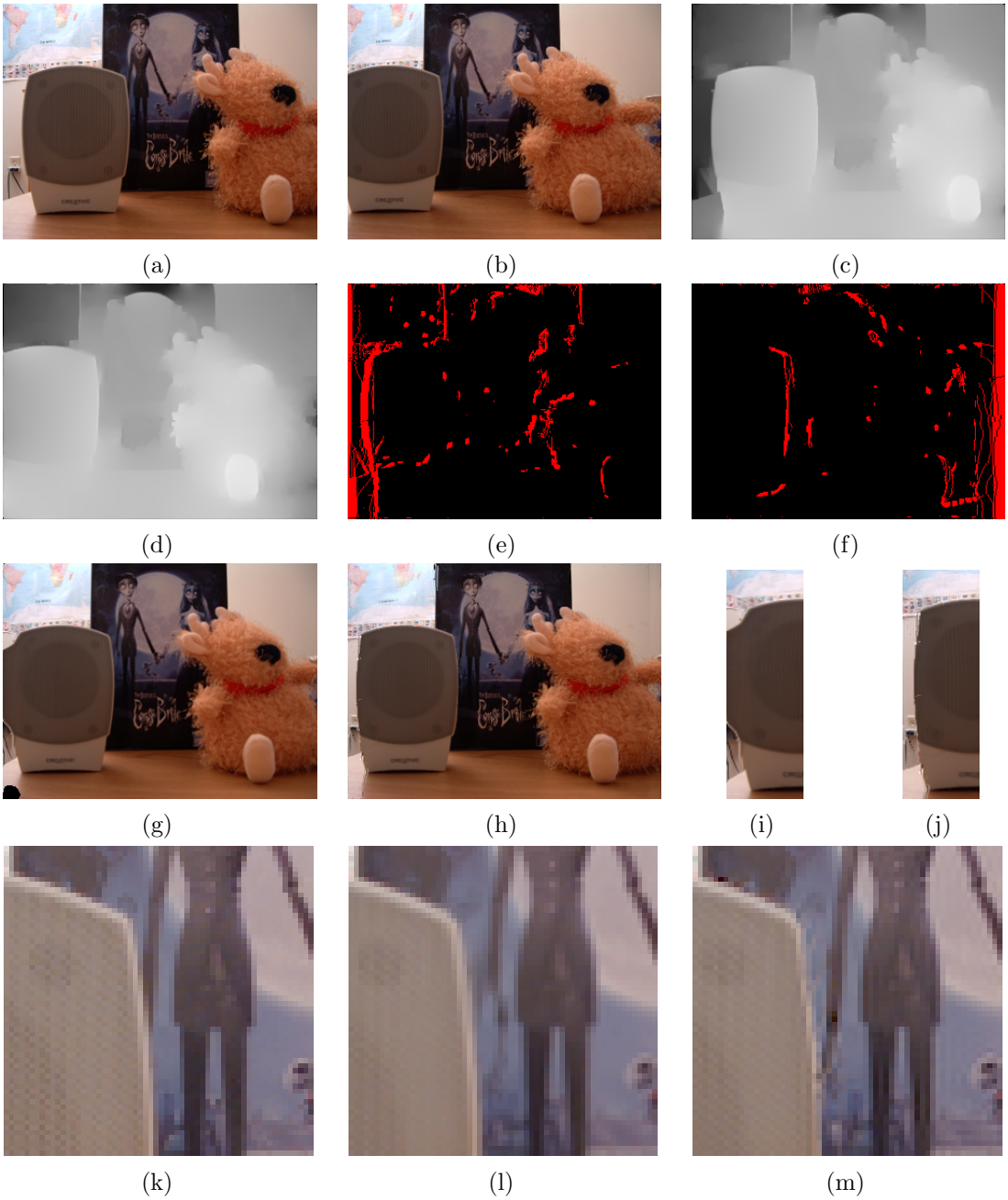
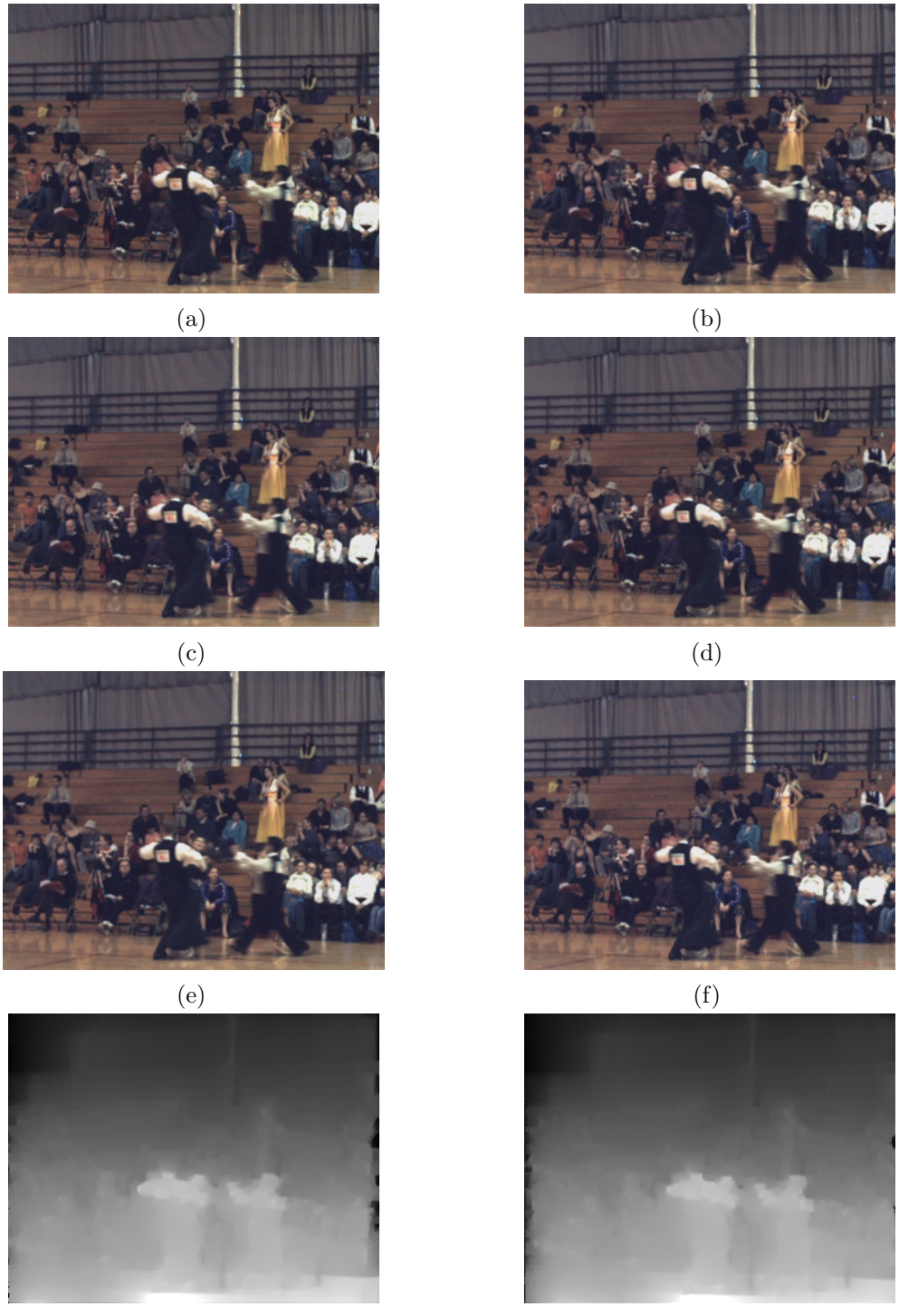


Figure 7·6: Original images: (a) left (I_L), and (b) right (I_R); estimated OF: (c) left-to-right, and (d) right-to-left ; pixels to be occluded in midpoint image J that come from: (e) left image, and (f) right image ; (g) pivoting-based reconstruction (h) spline-based reconstruction; (i) closeup of (g); (j) closeup of (h); closeups of (k) left image, (l) pivoting-based (m) spline-based reconstructions.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 7·7: Ballroom sequence: (a) left image at $\alpha = 0$; intermediate views at (b) $\alpha = 1/8$; (c) $\alpha = 3/8$; (d) $\alpha = 5/8$; (e) $\alpha = 7/8$; (f) right image at $\alpha = 1$; disparity maps of (g) left and (h) right images.

Chapter 8

Occlusion-aware view reconstruction using multiple input images

So far, we focused on view reconstruction and occlusion detection from two images only. In this chapter, we would like to extend our work to multiple input images. The main idea is that disparity and occlusion information can be more reliably estimated if there are more images available, i.e., what is not visible in two images is likely to be visible in other input images.

This chapter will start by briefly presenting how spline-based method can be easily extended to multiple images. Our main focus in this chapter, however, will be on improving pivoting-based methods by using additional images. We will propose a new method that adaptively estimates pivoted disparity by using multiple images. The method first estimates labels, and then uses these labels to direct a new variational formulation that chooses proper image pairs when computing disparity. The labels and computed disparities are then used in an adaptive reconstruction algorithm. The final reconstruction shows significant improvements over pivoting-based reconstruction that uses two images only.

8.1 Multi-view spline-based view reconstruction

An extension of method presented in Chapter 7 to multiple input images is trivial and requires no change in the method itself. As many images as available can be forward-compensated onto the intermediate view, thus creating additional intensity/color points.

In this case, multiple images will be beneficial to estimate more reliable disparities of available images. In Chapter 6, the proposed algorithm was able to estimate disparity of

visible areas and it solved the problem in occlusion areas by using a diffusion process. The proposed diffusion process essentially compensated for the lack of information and allowed us to intelligently guess disparity in occlusion areas.

On the other hand, if there are additional images, an optical flow algorithm can utilize these images to estimate disparity in occlusion areas instead of using diffusion.

There are many such works in the literature, for example Strecha and Van Gool (Strecha and Gool, 2002) proposed an optical flow method for N-images. This or other methods can be utilized to estimate more reliable disparities. Subsequently, these disparity fields can be used in multi-view spline-based reconstruction.

8.2 Multi-view pivoting-based view reconstruction

As we mentioned at the beginning of this chapter, we would like to focus on improving the pivoting-based approach. Let us first start by examining the problems in pivoting-based method.

8.2.1 Deficiencies of pivoting-based view reconstruction

1. *Absence of an underlying image in disparity estimation:* Pivoting-based method can be characterized as a backward-projected method (Section 4.1.1). The disparity of a pixel of the intermediate view is estimated by backward-projecting this point onto available images. This is in contrast to what we have proposed in spline-based method, where disparity of an input image is estimated first and then the intensities of the input image are forward-projected onto the intermediate view.

   Although this difference may seem to be insignificant, there is a crucial difference in estimating the disparity. In the forward-projected method, the disparity of a *known* image is estimated, therefore it is possible to use the underlying image for edge-preserving regularization purposes. As we have seen in Chapter 6, utilizing gradient information of the underlying image when estimating the disparity significantly improves the quality of the estimated disparity. Unfortunately, when estimating the

pivoted disparity, we have, obviously, no access to the intermediate view (that is what we are trying to estimate!).

Due to this problem, disparity estimated using the pivoted-based method is not of high accuracy, specifically the estimated disparity is usually excessively smooth, because there is no underlying image that will guide the regularization. An example pivoted disparity was previously shown in Fig. 4·4.c. Although it is possible to utilize robust statistics in the regularization term, as in the work of Black and Anandan (Black and Anandan, 1996), edge-preserving regularization leads to sharper disparity maps because discontinuities coincide with object boundaries.

2. *Insufficient information to estimate occlusion areas:* Given only a single disparity field pivoted on the intermediate view, it is unclear how to estimate occlusion areas. For example, in Chapter 6, we measured the compatibility of left-to-right and right-to-left disparity fields, however, in the pivoting-based method, one can estimate a single disparity field, therefore no such compatibility measurement can be done. Thus, we have no clear mechanism of finding occlusion areas in pivoting-based method.

   In fact, even if we were able to estimate occlusion areas on the intermediate view, handling of occlusion areas would be problematic as well. For example, we proposed an image-driven disparity inpainting method in Chapter 5. However, we cannot use such a method to handle occlusions because, yet again, we have no underlying image to guide the inpainting.

In the following sections, we will propose methods to solve both problems. A coarse intermediate image will be used to solve the first problem and multiple input images will be used to solve the occlusion problem.

8.2.2 Edge-preserving regularization using coarse intermediate image

In this section, we will solve the first deficiency mentioned in the previous section. As demonstrated in Chapter 6, edge-preserving (anisotropic) regularization preserves disparity

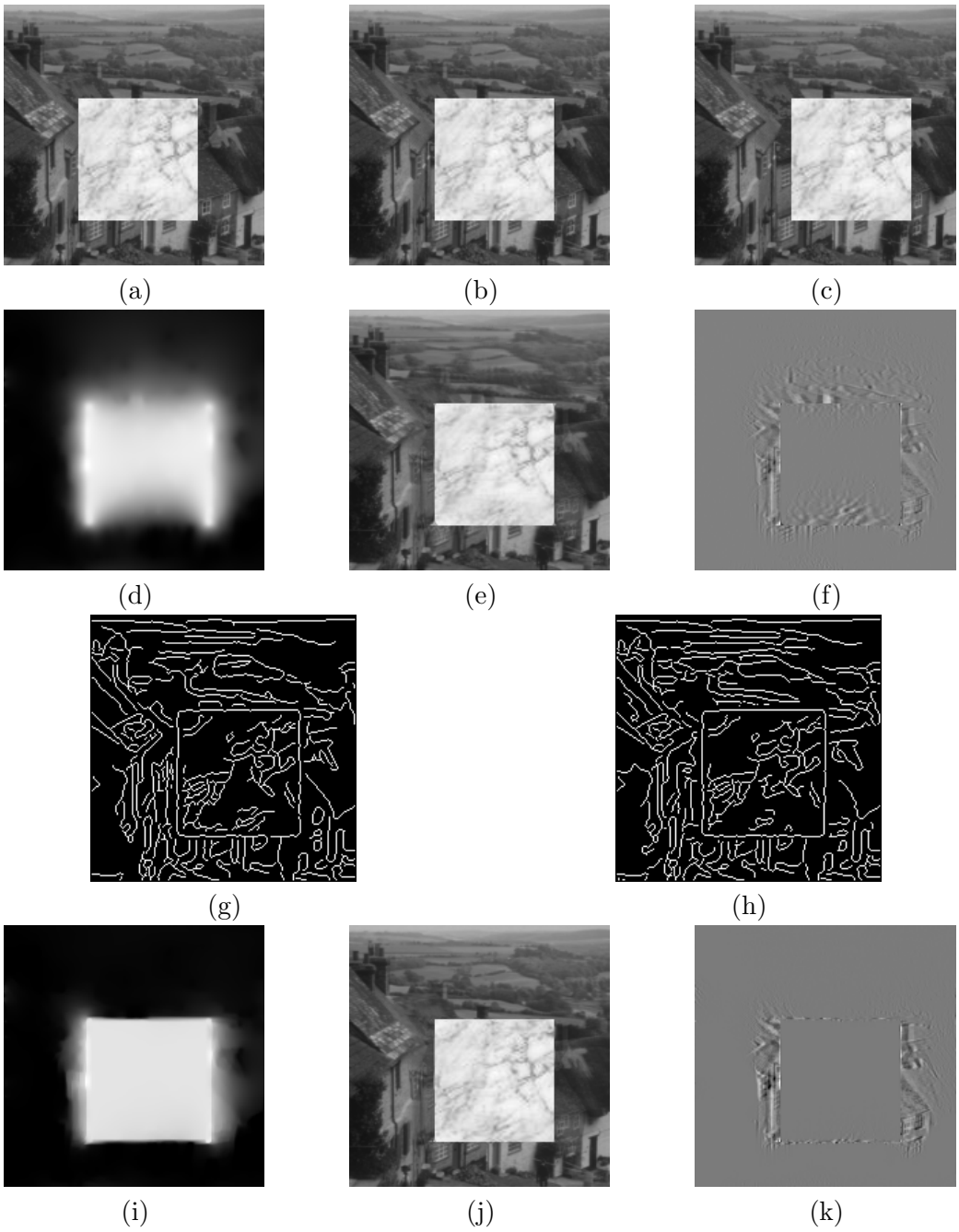


Figure 8·1: Experimental results for a synthetic image. Original (a) left (b) intermediate and (c) right images; (d) pivoted-disparity estimated using isotropic diffusion; (e) intermediate image reconstructed using pivoted disparity from (d); (f) prediction error; edge map of (g) true intermediate view and (h) reconstructed intermediate image; (i) pivoted-disparity estimated using anisotropic diffusion; (j) intermediate image reconstructed using images in (a), (c) and pivoted disparity in (i); (k) prediction error.

edges better than isotropic diffusion, but requires an image gradient to guide the diffusion process. The main difficulty is in providing such a gradient.

The pivoting-based method using two images (as described in Section 4.1.1) results in intermediate views which have distorted texture but, surprisingly, the edge information is usually reliable. Although this seems to be counter-intuitive, the reason is that visible edges are matched easily and edges near occlusion areas belong to the object closer to the camera, therefore its edges are visible as well. Figure 8·1 shows an experimental result proving the point. The images shown in Fig. 8·1.a and c are the input left and right images and Fig. 8·1.b is the true intermediate image. The estimated disparity without using edge-preserving diffusion is shown in Fig. 8·1.d. It is computed *via* the following minimization:

$$\arg\min_{\mathbf{d}(\mathbf{x})} \int\int_{\mathbf{x}\in\Omega_J} \left(I_L(\mathbf{x}-\alpha\mathbf{d}(\mathbf{x})) - I_R(\mathbf{x}+(1-\alpha)\mathbf{d}(\mathbf{x}))\right)^2 + \lambda\left(||\nabla u||^2 + ||\nabla v||^2\right) d\mathbf{x} \quad (8.1)$$

where u and v are the horizontal and the vertical components of disparity. Clearly the resulting disparity is excessively smooth. The reason for smoothness is the isotropic diffusion in the formulation.

The image in Fig. 8·1.e shows the reconstructed image using this disparity. Although there are gross prediction errors (shown in Fig. 8·1.f), the edge maps obtained using Canny edge detector from the true and reconstructed intermediate images (Fig. 8·1.g and h) are very similar.

Therefore, we propose to use the coarse intermediate image shown in Fig. 8·1.e to guide the edge-preserving diffusion as follows:

$$\arg\min_{\mathbf{d}(\mathbf{x})} \int\int_{\mathbf{x}\in\Omega_J} \left(I_L(\mathbf{x}-\alpha\mathbf{d}(\mathbf{x})) - I_R(\mathbf{x}+(1-\alpha)\mathbf{d}(\mathbf{x}))\right)^2 + \lambda\left(F_{\mathbf{x}}(u, J_c) + F_{\mathbf{x}}(v, J_c)\right) d\mathbf{x}, \quad (8.2)$$

where J_c is the coarse intermediate image and $F_{\mathbf{x}}$ is edge-preserving diffusion term defined in (5.5). The difference between (8.1) and (8.2) is that (8.2) achieves edge-preserving diffusion using J_c.

The disparity shown in Fig. 8·1.i is computed by minimizing (8.2). It is clear that the object shape is very well-preserved. The intermediate view obtained using this disparity field and its prediction error are shown in Fig. 8·1.j and k respectively. As is clear from the prediction error, the distortions on the top and the bottom of the object in Fig. 8·1.f are highly reduced because the excessive smoothness of disparity field is eliminated. The reason is that although these areas are not occlusion areas, due to isotropic regularization, they were previously assigned incorrect disparity values in (8.1). Edge-preserving regularization in (8.2) eliminated this problem and these areas are now assigned accurate disparity values, thus are properly reconstructed.

To summarize this section, by proposing to use a coarse image for disparity estimation, we solved the first deficiency mentioned in Section 8.2.1. However, in the experimental result, we see that areas on the left and right of the object are still problematic because these areas are occlusion areas and edge-preserving regularization is not sufficient to solve the occlusion problem. The problem is due to insufficient information; a point is visible only in one of the images, thus no matching is possible. Therefore, next, we propose to use additional images to solve the problems in these areas. The idea is that given multiple images, these areas are expected to be visible in at least two of the input images, therefore, can be matched to compute an accurate disparity value.

8.2.3 Utilizing multiple images in occlusion areas

In this section, our aim is to solve the second deficiency mentioned in Section 8.2.1 by using multiple images. Let us discuss how using multiple images will improve the disparity estimation in occlusion areas.

Without losing generality, let us consider four input images as shown in Fig. 8·2. Although simple, this figure can successfully convey the idea of using multiple images. Images and their cross-sections are shown at the top and bottom, respectively. We would like to reconstruct the intermediate image J using input images I_1, I_2, I_3 and I_4. Note that areas A and B are being occluded/exposed between the four images.

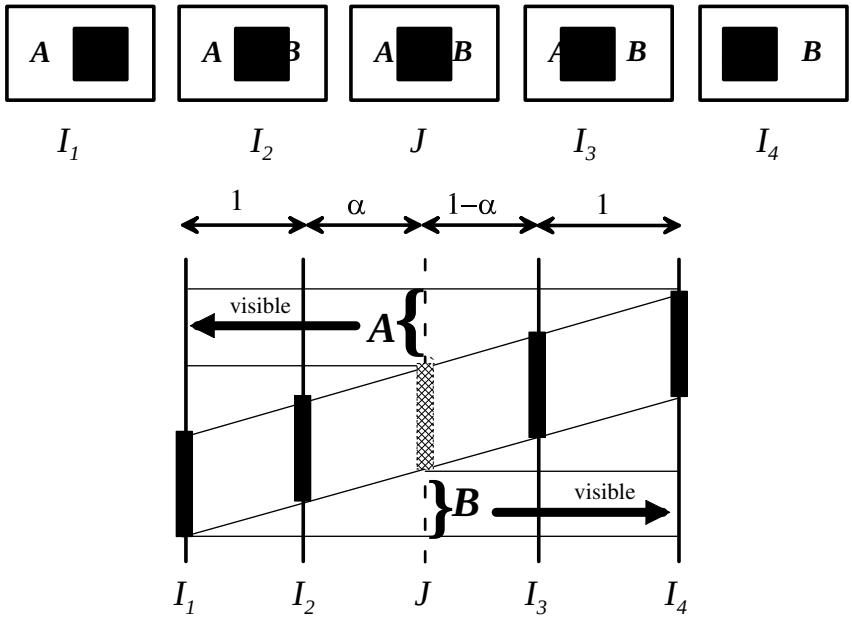


Figure 8·2: Using four images is sufficient for multi-view pivoting based reconstruction. Occlusion areas A and B (shown in images above and cross-sections below) can be estimated either from (I_1, I_2) or (I_3, I_4). All other points that do not belong to A or B can be estimated from (I_2, I_3).

In the case of pivoting that uses two input images, I_2 and I_3 would be the input images and a disparity field pivoted on J would be estimated. For most points of J, it is possible to estimate accurate disparity values because they are visible in both I_2 and I_3. However, areas A and B are occluded between the images, therefore it is not possible to estimate disparities for these areas.

If there are additional images to the left and right of I_2 and I_3, then areas A and B would be visible in at least two images. Then, it should be possible to estimate the disparity of area A using I_1 and I_2 and disparity of area B using I_3 and I_4. Therefore, we need a formulation that will estimate disparity of J by choosing among three pairs: (I_1, I_2), (I_3, I_4) or (I_2, I_3).

8.2.4 Estimation of labels

In order to implement the idea of switching between image pairs from the previous section, obviously we need to first find where areas A and B are. We propose to use the method developed in Section 5.3 for this purpose:

It is possible to estimate a disparity field $\mathbf{d}_{12}$ from I_1 to I_2 pivoted on I_1. The method proposed in Section 5.3 will yield the area B by using $(1+\alpha)\mathbf{d}_{12}$. The coefficient $(1+\alpha)$ is used to normalize the disparity field so that it is correctly mapped onto J. The estimated area B is exposed between I_1 and I_2, and therefore is visible in I_3 and I_4.

Similarly a disparity field $\mathbf{d}_{43}$ from I_4 to I_3 pivoted on I_4 can be estimated. Using $(2-\alpha)\mathbf{d}_{43}$ will yield area A, which is visible in I_1 and I_2. Therefore, by using $\mathbf{d}_{12}$ and $\mathbf{d}_{34}$, we can find the labels of points in J: visible, occluded, exposed.

Now that we have the labels, in the next section we will propose a new variational formulation that utilizes these labels.

8.2.5 Proposed variational formulation

Let $I_1 : \Omega_1 \rightarrow R^+$, $I_2 : \Omega_2 \rightarrow R^+$, $I_3 : \Omega_3 \rightarrow R^+$, $I_4 : \Omega_4 \rightarrow R^+$ be input images, and let $J : \Omega_J \rightarrow R^+$ be the intermediate image to be reconstructed, let $\mathbf{x} \in \Omega_J$. We would like to compute a disparity field: $\{\mathbf{d}(\mathbf{x}) = [u(\mathbf{x}), v(\mathbf{x})]^T\}_{\mathbf{x}\in\Omega_J}$ and we would like the formulation be directed by a label field $L : \Omega_L \rightarrow R^+$.

Let us first propose three sets of prediction errors as follows:

$$\theta_{12}(\mathbf{x}) = I_1(\mathbf{x} - (1+\alpha)\,\mathbf{d}(\mathbf{x})) - I_2(\mathbf{x} - \alpha\,\mathbf{d}(\mathbf{x})), \tag{8.3}$$

$$\theta_{23}(\mathbf{x}) = I_2(\mathbf{x} - \alpha\,\mathbf{d}(\mathbf{x})) - I_3(\mathbf{x} + (1-\alpha)\,\mathbf{d}(\mathbf{x})), \tag{8.4}$$

$$\theta_{34}(\mathbf{x}) = I_3(\mathbf{x} + (1-\alpha)\,\mathbf{d}(\mathbf{x})) - I_4(\mathbf{x} + (2-\alpha)\,\mathbf{d}(\mathbf{x})). \tag{8.5}$$

$\theta_{12}, \theta_{23}, \theta_{34}$ measure prediction errors for image pairs (I_1, I_2), (I_2, I_3) and (I_3, I_4), respectively. The $(1-\alpha), (1+\alpha), (2-\alpha)$ coefficients normalize the disparity vectors depending on the distance between images.

For every $\mathbf{x} \in \Omega_J$, it is possible to evaluate these prediction errors. Most points, except points in the occlusion areas in A and B, will yield small prediction errors in all three cases. However, in the occlusion areas only one of them will yield a small prediction error. Specifically, in area A, θ_{12} will be small whereas in area B, θ_{34} will be small.

Let us define a label field $L(\mathbf{x})$ as follows: if $L(\mathbf{x})$ is -1, 0 or 1, we would like θ_{12}, θ_{23} and θ_{34} be used as prediction errors, respectively[1]. Combining these labels and prediction errors we propose three prediction errors to be used in our formulation as follows:

$$P_{12}(\mathbf{x}) = \delta(L(\mathbf{x})+1)(\theta_{12}(\mathbf{x}))^2, \tag{8.6}$$

$$P_{23}(\mathbf{x}) = \delta(L(\mathbf{x}))(\theta_{23}(\mathbf{x}))^2, \tag{8.7}$$

$$P_{34}(\mathbf{x}) = \delta(L(\mathbf{x})-1)(\theta_{34}(\mathbf{x}))^2, \tag{8.8}$$

and then combine them in a single cost term as follows:

$$e_P(\mathbf{x}) = P_{12}(\mathbf{x}) + P_{23}(\mathbf{x}) + P_{34}(\mathbf{x}), \tag{8.9}$$

where $\delta(x)$ is the Kronecker delta function. Since this function is not continuous, we propose to use an approximation such as $\delta(x) = \lim_{k\to\infty} e^{-kx^2}$. For example, $k = 10^{10}$ yields a very good approximation.

Clearly, e_P adaptively selects different pairs of input images depending on the labels, L . For example, if $L(x) = -1$, then P_{12} is used because, $\delta(L(\mathbf{x})+1) = 1$ and $\delta(L(\mathbf{x})) = \delta(L(\mathbf{x})-1) = 0$.

Now that we have a prediction term, we propose to use the following smoothness term, which will enforce edge-preserving diffusion:

$$e_S(\mathbf{x}) = F_{\mathbf{x}}(u_L, J_c) + F_{\mathbf{x}}(v_L, J_c), \tag{8.10}$$

with $F_{\mathbf{x}}$ defined in (5.5) and J_c being coarse intermediate image reconstructed using

[1]The values of labels have no significance and are chosen randomly. It is possible to use other values for labels such as 1, 2 and 3.

isotropic disparity estimation as proposed in 8.2.2.

Combining these terms we propose to minimize the following energy with respect to $\mathbf{d}$:

$$E = \int\int_{\mathbf{x}\in\Omega_J} e_P(\mathbf{x}) + \lambda e_S(\mathbf{x}) d\mathbf{x} \tag{8.11}$$

Derivation of Euler Lagrange equations is given in Appendix C.

Once a disparity field is estimated, it is possible to reconstruct $J(\mathbf{x})$ by using any intensity value along the disparity vector, but weighted averaging, or simply averaging of intensities lead to better results. Considering this, we propose to reconstruct the intermediate view as follows:

$$J(\mathbf{x}) = \delta(L(\mathbf{x})+1)\,\xi_{12} + \delta(L(\mathbf{x}))\,\xi_{23} + \delta(L(\mathbf{x})-1)\,\xi_{34} \quad \forall \mathbf{x} \in \Omega_J, \tag{8.12}$$

where L is the estimated label field and $\xi_{\cdot}$ are the intensity averages along disparity vector $\mathbf{d}(\mathbf{x})$ defined as follows:

$$\begin{aligned}
\xi_{12} &= \frac{1}{2}[I_1(\mathbf{x} - (1+\alpha)\,\mathbf{d}(\mathbf{x})) + I_2(\mathbf{x} - \alpha\,\mathbf{d}(\mathbf{x}))], \\
\xi_{23} &= \frac{1}{2}[I_2(\mathbf{x} - \alpha\,\mathbf{d}(\mathbf{x})) - I_3(\mathbf{x} + (1-\alpha)\,\mathbf{d}(\mathbf{x}))], \\
\xi_{34} &= \frac{1}{2}[I_3(\mathbf{x} + (1-\alpha)\,\mathbf{d}(\mathbf{x})) - I_4(\mathbf{x} + (2-\alpha)\,\mathbf{d}(\mathbf{x}))].
\end{aligned} \tag{8.13}$$

Note that for every $J(\mathbf{x})$, only one of the values in (8.13) is used in (8.12) because of the $\delta(\cdot)$ terms.

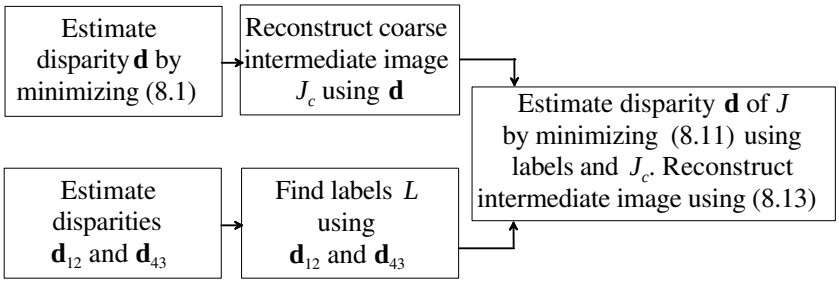


Figure 8·3: The steps of the proposed occlusion-aware pivoting-based multiview intermediate view reconstruction method.

Table 8.1: PSNR values of intermediate views with different reconstruction methods for synthetic test sequence.

Description of method	PSNR
Pivoting-based method using two images	30.77dB
Pivoting-based method using two images and a coarse intermediate image	32.84dB
Proposed method	34.15dB

To summarize the proposed method, we show the steps in Fig. 8·3. The proposed formulation solves both deficiencies of pivoting-based method mentioned in Section 8.2.1. We will show the efficacy of the proposed method on synthetic and real-world images in the next section.

8.2.6 Experimental results

We generated two additional images for the synthetic test sequence shown in Fig. 8·1. Four input images are shown in Fig. 8·4.a-d. True disparity, intermediate image and label map are shown in Fig. 8·4.e-g. The labels estimated using the method proposed in Section 5.3 are shown in Fig. 8·4.h. Black, gray and white colors indicate that (I_1, I_2), (I_2, I_3) and (I_3, I_4) image pairs should be used in these areas, respectively. Although there are false positives on top and bottom of the object, since these areas are visible in all images, they can be predicted from any pair.

Disparity estimated using the formulation proposed Section 8.2.5 is shown in Fig. 8·4.i. When compared to Fig. 8·1.i, which used only two images, the improvement in occlusion areas is clear. The intermediate image reconstructed using this disparity and label field from Fig. 8·4.h is shown in Fig. 8·4.j. The prediction error in Fig. 8·4.k clearly shows the improvement. Numerical results of different methods are shown in Table 8.1. Proposed edge-preserving diffusion improved results of pivoting-based method that uses two images by more than 2dB, and proposed multiview method improved this result by another 1dB.

We also tested the proposed method on natural images. We used four frames (10^{th}, 16^{th}, 22^{nd}, 28^{th} frames) of *Flowergarden* sequence to reconstruct the 19^{th} frame. The four

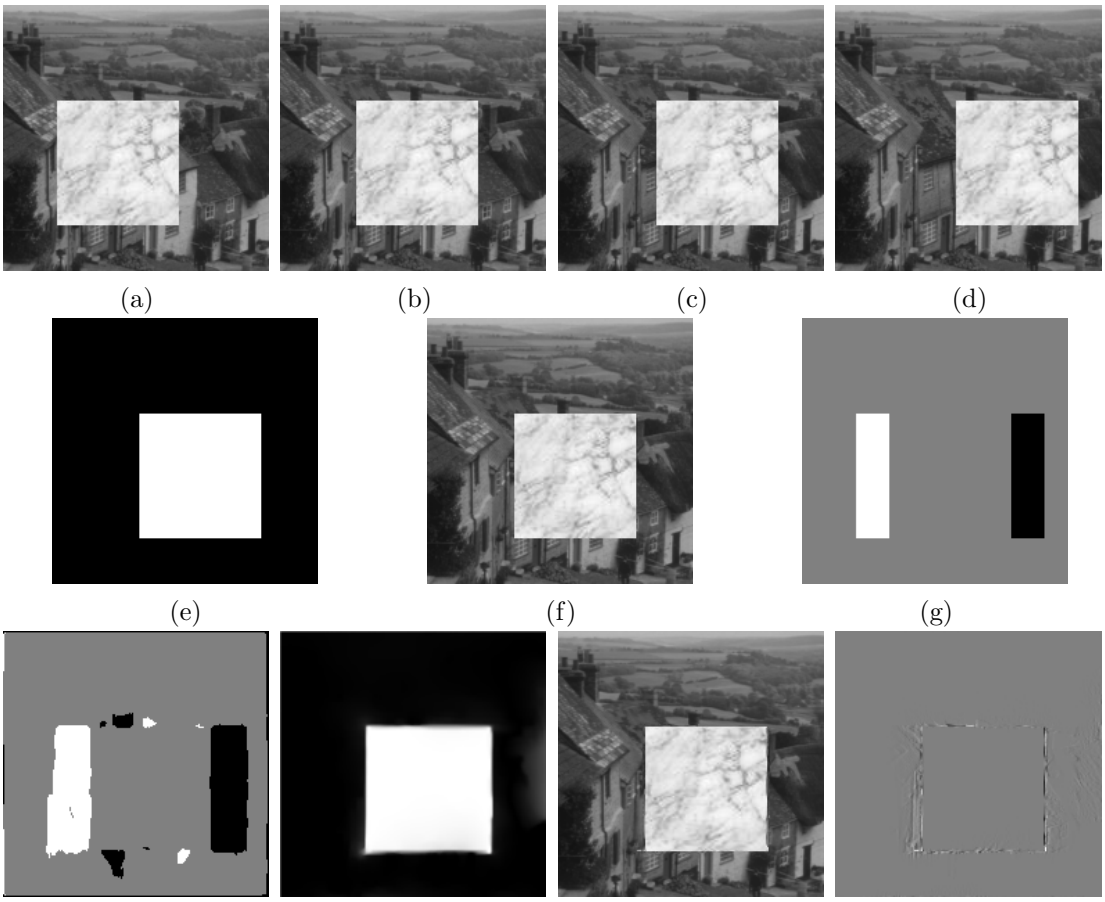


Figure 8·4: Experimental results for a synthetic sequence. (a) I_1, (b) I_2, (c) I_3, (d) I_4; ground truth (e) disparity, (f) intermediate image, (g) label map; (h) estimated labels (black, gray and white indicate (I_1, I_2), (I_2, I_3) and (I_3, I_4) image pairs to be used) (i) estimated disparity by minimizing (8.11); (j) reconstructed intermediate image; (k) prediction error.

original images are shown in Fig. 8·5.a-d. It can be noticed that the tree trunk occludes the house in the background. The disparity estimated using pivoting-based method that uses two images, i.e., 16^{th} and 22^{nd} frames, is shown in Fig. 8·5.e. It is excessively smooth. The reconstruction using this disparity field and the two input images is shown in Fig. 8·5.f. The occlusion area is poorly reconstructed; the texture around the tree trunk is highly distorted (closeup in Fig. 8·5.k).

The label map that is estimated using the method proposed in Section 8.2.4 is shown in Fig. 8·5.g. Black areas are to be predicted from frames (#22, #28), white areas from (#10,

#16), and gray areas from (#16, #22). Estimated disparity using the method proposed in Section 8.2.5 is shown Fig. 8·5.h. When compared to the previous result in Fig. 8·5.e, the new disparity has object boundaries. Reconstructed intermediate view using (8.12) is shown in Fig. 8·5.i. Since the input sequence is a video, we can subjectively compare the reconstructed view to the 19^{th} frame of the sequence. Closeup of the original 19^{th} frame and reconstructions are shown in Fig. 8·5.j-l. Although there are slight shifts, texture in the background is clearly distinguished in new reconstruction and very similar to the original frame. For example, the windows of the house cannot be identified in Fig. 8·5.k, while they are easily identified in Fig. 8·5.l. Similarly, the tree branches behind the house are distorted in Fig. 8·5.k, but are better reconstructed in Fig. 8·5.l.

8.3 Why not estimate labels and disparity simultaneously?

The method in the previous section, although successful, is composed of a few steps: first estimation of labels and coarse intermediate image, followed by estimation of disparity using these labels and the coarse image.

We explored the possibility of a new cost function that simultaneously estimates labels *and* disparity without using a coarse image. Our aim was to propose a joint formulation that estimates all unknowns simultaneously. Specifically, we formulated the joint problem as the following minimization:

$$E = \int\int_{\mathbf{x}\in\Omega_J} \underbrace{\Phi(d_{12},d_J)\theta_{12}(\mathbf{x})^2 + \Phi(d_{23},d_J)\theta_{23}(\mathbf{x})^2 + \Phi(d_{34},d_J)\theta_{34}(\mathbf{x})^2}_{\text{Prediction terms}} + \underbrace{\lambda_1\Psi(\mathbf{d},J_c)}_{\text{Regularization term}} + \underbrace{\lambda_2(1-\Phi(d_{12},d_J)) + \lambda_3(1-\Phi(d_{23},d_J)) + \lambda_4(1-\Phi(d_{34},d_J))}_{\text{Occlusion priors}}. \quad (8.14)$$

The formulation is similar to the formulation of the occlusion-aware optical flow. The idea is to disable the prediction terms, θs, that are not reliable. Specifically, $\theta_\cdot$ are individual prediction errors as given in (8.5). $\Phi(\cdot,\mathbf{d}_J)$ functions measure the reliability of $\mathbf{d}_J$, by

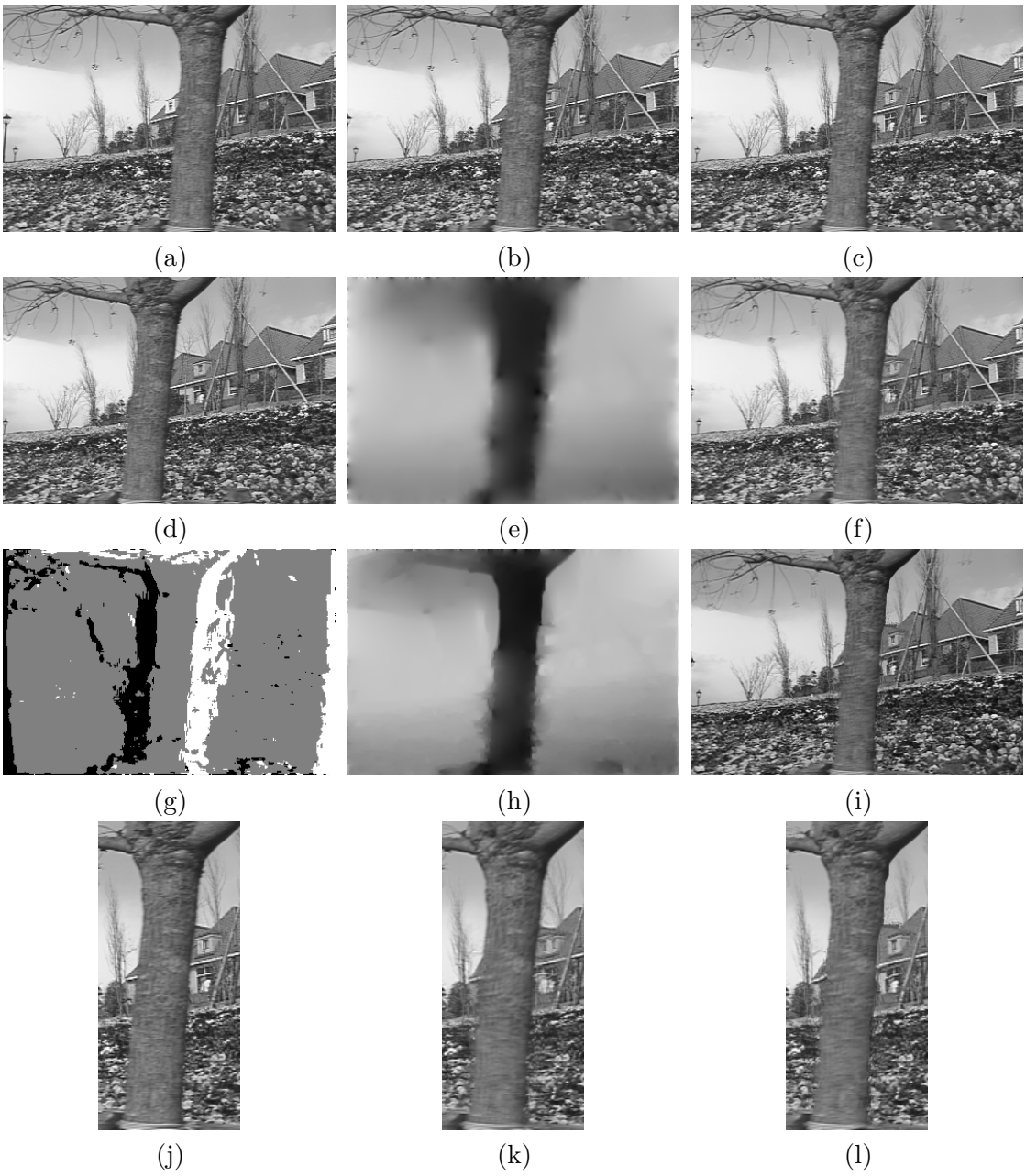


Figure 8·5: Original frames (a) #10, (b) #16, (c) #22, (d) #28; (e) estimated disparity using pivoting-based method that uses two-images; (f) reconstructed intermediate view by using disparity shown in (e); (g) estimated label map (white: frames (10,16), black: frames (22,28), gray: frames (16,22)); (h) estimated disparity using the proposed variational approach; (i) reconstructed intermediate view using proposed approach; (j) closeup of true frame #19; (k) closeup of the result of the method that uses two-images; (l) closeup of the result of proposed approach.

comparing to the other disparity fields, similar to the $D(\cdot)$ function in (6.9). If the vector field $\mathbf{d}_J$ is estimated to reliable, then $\Phi(\cdot)$ will be close to one, therefore allowing the use of corresponding prediction term. If $\mathbf{d}_J$ is not reliable then Φ will disable the prediction term because $\Phi(\cdot)$ will be close to zero.

Unfortunately, this method offered limited improvement. Upon investigating, we concluded with the following reasons for limited performance:

1. *Ambiguity*: If the pivoted location is not in between available images, there are at least two disparity values that satisfy the prediction error which leads to an ambiguous solution. Consider Fig. 8·6, where the intermediate image is positioned outside of input images. Without losing generality, let us assume that the background is static. We would like a disparity vector to be pivoted on J and pass through I_1 and I_2, and minimize θ_{12}. The ambiguity is that since we do not have the underlying image, any solution that minimizes prediction error and satisfies the smoothness constraint will be an acceptable solution. In this case, for the upper part of the object, there are two solutions $||\mathbf{d}|| = 0$ (which matches background that is static) and $||\mathbf{d}|| \neq 0$ (which matches the object in I_1 and I_2). Both of these solutions would reasonably satisfy the smoothness constraint as well, because these areas are near the object boundary and are affected by disparity values in both object and background.

   Note that this ambiguity was not present in the method proposed in previous section, because precomputed labels explicitly (and correctly) directed the solution method.

2. *Complex formulation*: The formulation has several terms which should be weighed properly. Our experimental results show that there is usually not a single optimal set of parameters that would work with any data and adjusting parameters for each set is not practical at all. Therefore this problem rendered the formulation unpractical.

3. *Lack of anisotropic diffusion*: Yet, again, the lack of anisotropic diffusion led to unsatisfactory results. Although, as we proposed in the previous section, one can use

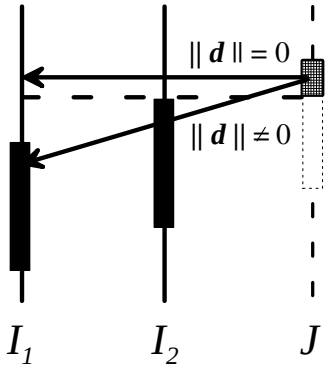


Figure 8·6: Illustration of ambiguity when disparity is not pivoted between the input images. Shaded area in J can be assigned two disparity values that minimize prediction error.

a coarse intermediate image for diffusion purposes, the other difficulties made a joint approach unrealistic.

8.4 Conclusions

In this chapter, we first discussed how the method proposed in Chapter 7 can be extended to multiple views. Next, we pointed out the limitations of pivoting-based disparity estimation, specifically the absence of an underlying image (thus the lack of edge-preserving regularization), and its inability to handle occlusion areas. Next, we proved that although pivoting-based reconstruction using two images creates distorted texture in intermediate view, it reconstructs reasonable edge information. Exploiting this fact, we proposed to use a coarse intermediate image in disparity estimation for edge-preserving regularization purposes. Then, we proposed a new variational pivoting-based method that works on multiple images. The basic idea is that when multiple images are available, a point in the intermediate image is visible in at least two images. The formulation uses this fact to choose proper image pairs and estimate disparity vectors. The selection process is guided by labels computed using the method proposed in Section 5.3. The results show significant improvements over pivoting-based method that uses two images only.

Finally, let us point the difference between the method proposed in this chapter and in

Chapter 6. In Chapter 6, since we had only two input images, the estimation of disparity in occlusion areas was impossible. However, by using available disparities of visible points, we intelligently guessed the disparity of occlusion areas *via* a diffusion process; and effectively compensated for the lack of information. On the other hand, in this chapter, we utilized additional images that enabled us to estimate reliable disparities in the occlusion areas for all points.

Chapter 9

Applications of proposed methods

This chapter presents several applications of methods presented in this dissertation. The application areas are health care, personal communication, video compression and space exploration.

9.1 Health care: Virtual reality for bedridden patients

In collaboration with NeuroMuscular Research Center at Boston University, we worked on building an exercise bed for bedridden patients that will help to recover their balance after recovery from illnesses (Oddsson et al., 2007; Oddsson et al., 2006). Our part in the project was to utilize 3D displays and proposed view reconstruction algorithm to create a virtual reality environment for a patient in bed.

9.1.1 Introduction

3D visualization equipment finds various application areas; we have mentioned some of these applications in Chapter 1. Mainly, these applications focus on creating a better representation of medical data so that medical personnel can improve their diagnosis. These applications rarely utilize 3D tools to help patients with decreasing their recovery time.

Recently, we collaborated with NeuroMuscular Research Center at Boston University to create a system that utilizes 3D displays and head-mounted 3D glasses to create a virtual reality environment. The system focuses on patients that had to stay in bed for an extended time period due to a spinal cord injury or neurological diseases. The results of neuromuscular research indicates that having stayed in bed for such a long time, these patients have difficulties in maintaining their balance when standing, even after they re-

Figure 9·1: The prototype bed. Treadmill shown on the left acts as ground and promotes walking. Computer is used to control 3D monitors positioned on the left and above the bed.

covered from illnesses.

Moreover, previous research shows that if the patients are trained while they are still in bed, so that they experience the gravitational force as if they were standing on their own, then patient's recovery time of motor functions reduces significantly.

Considering this potential, Oddsson et al. (Oddsson et al., 2004) created a training system in a 90-degree tilted room. The room contains objects that would be in an ordinary room, such as a table and a chair, but they are mounted on the wall to create a standing feeling when lying on a mat on the floor. The subject wears a harness that pulls him/her toward the wall (which is perceived as the ground) therefore also delivering the sense of gravity.

Although encouraging, the system was obviously non-portable. Considering the severity of injuries of patients, it is difficult to see this system being used in any hospital. In order to solve this issue, they proposed to build a portable system, that can be easily transported to patients instead of transporting patients to the system.

9.1.2 System design

The system is mounted on an ordinary hospital bed as shown in Fig. 9·1. In order to create a similar feeling as the one delivered by the tilted room, we utilized two automultipscopic 3D displays from Stereographics Corp. namely SynthaGram SG222, with a resolution of 3840 $\times$ 2400 pixels and SynthaGram SG202 with a resolution of 1600$\times$1200 pixels. The high resolution and quality SG222 is directly in front of the subject. The other display is placed on the side of the subject since a user will spend less time looking to sides; this display is enough to provide the desired effect.

These 3D displays are intented to be virtual "windows" to an outside environment. The images displayed on these screens promote a visually-induced reorientation illusion. Since patients are able to "look out of a window" they, in fact, experience the sense of being upright with respect to gravity.

Both of the these displays require nine individual views as input. Since we do not have nine individual cameras, we captured two (rarely three) views of a scene and used the proposed view reconstruction algorithm to generate additional views. The algorithm is further enhanced by using anti-aliasing filters (Konrad and Agniel, 2006).

In the first stage of the project, images around Boston University were captured using a stereo camera setup that is composed of two Olympus digital still cameras. To further improve the effectiveness of the system, the system will be tailored to each patient by capturing images that are familiar to him/her such as his/her own backyard or living room. The setup and software for this step have been completed and applied in practical experiments.

This specific application demonstrates the versatility of methods proposed in this dissertation. The 2-camera setup can easily be carried by health care professionals to the home of a patient. Such a professional, very likely unskilled in camera calibration and other technical details, will easily capture a stereo pair. Later, this pair will be displayed on the automultiscopic display with little effort by using the proposed view reconstruction method.

Finally, as an alternative to automultipscopic 3D displays, we utilized a head-mounted stereo display (HMSD), *3DVisor*. This specific HMSD is able to deliver a virtual reality environment given a 3D model. We designed virtual rooms using a 3D modeling software *Blender*. When a subject wears this HMSD, he/she perceives that he/she is standing in a room upright and interacts with objects surrounding him/her. This alternative method will be more cost-effective as HMSDs are substantially cheaper than automultiscopic displays. However, the subject may not be as comfortable as in the case of 3D displays because of the interocular *vs.* intercamera distance problem as we mentioned in Chapter 2.

9.1.3 Experimental results

Experimental results on healthy subjects show that the strength and balance functions improved significantly. More details can be found in (Oddsson et al., 2007; Oddsson et al., 2006). The system will soon be used on patients at Boston Medical Center.

9.1.4 Conclusions

We presented a system where 3D displays and view reconstruction algorithms are used for a health care application. The system can be used by any medical professional without learning technical details of 3D vision. This system illustrates a very good example of application of our work.

9.2 Personal communications: Frame rate conversion to enhance videos captured by mobile phones

This application demonstrates that view reconstruction algorithm used along time axis in monoscopic video sequences can increase frame rate of low-quality videos. Specifically, we use the proposed view reconstruction algorithm to generate 30 frames-per-second (fps) videos from 5-6fps mobile phone videos.

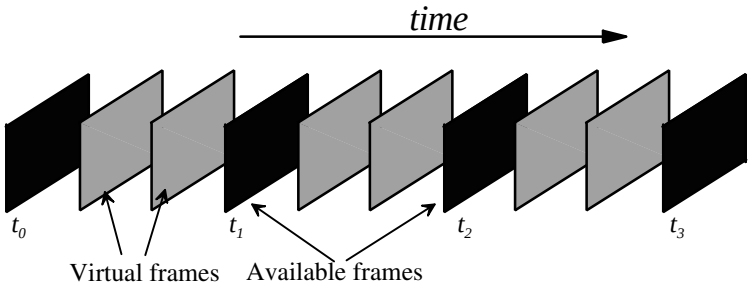


Figure 9·2: Virtual frames can be reconstructed from available frames.

9.2.1 Introduction

So far, we focused on intermediate view reconstruction in stereo or multi-camera setups. However, view generation can be used to increase the frame rate of a monoscopic video sequence. This can be achieved by applying view reconstruction along time in monoscopic video sequences (Fig. 9·2). The disparity estimation step in the reconstruction algorithm is replaced by motion estimation. We use the optical flow algorithm proposed in Chapter 6 for motion estimation, as it can be used with both stereo and video data.

The main application of this work is the enhancement of video sequences captured by a mobile phone. Since mobile phones have limited processing power, they typically capture 5-6fps. This rate is insufficient when compared to 30fps which is the required minimum rate for a smooth playback of a video sequence. Therefore, video sequences captured by mobile phones are perceived as 'jerky', i.e., object motion is perceived as unnatural.

When creating a system for view generation in video sequences we exploit the poor human perception when watching a video. Namely, we cannot perceive slight texture distortions in a video sequence since it is shown on the screen for a fraction of a second. Moreover, unless there is a very fast moving object in the scene, the occlusion effects are minimal between frames. Therefore, as we will show in experimental results, we can reconstruct intermediate views by skipping the occlusion handling step (i.e., Step #4 in Chapter 7).

We should note that we assume that the frames are captured in equal time intervals. In rare cases, the processor of the phone cannot keep up with the amount of motion in the

scene due to the time spent on motion estimation during compression. In these cases, the frame rate drops below the average 5-6 fps. Yet, even when we used such very low frame rate videos as input, the reconstructed video sequences were still perceived as smooth.

9.2.2 Proposed method

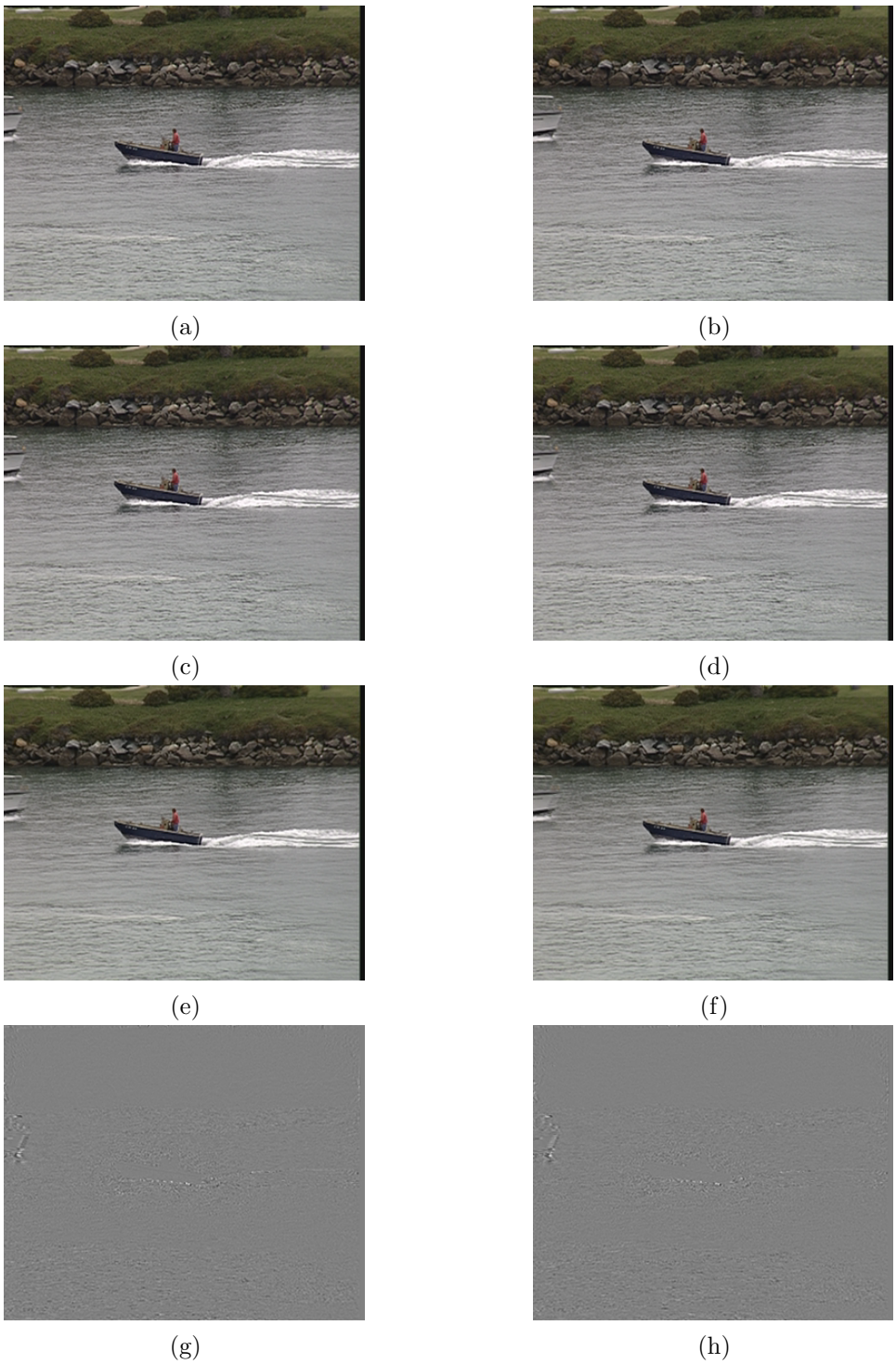
We utilize the view reconstruction algorithm presented in Chapter 7. We estimate the vector field between input frames using occlusion-aware optical flow estimation (Chapter 6) and use spline-based reconstruction (Chapter 4). The important fact is that we no longer focus on disparity therefore the occlusion-aware optical flow method successfully handles vertical disparities as well. This is an important difference of our method from previous work as it can successfully handle disparity as well as 2D motion.

Another additional quality of our approach is that spline-based reconstruction needs only one set of vector fields (left-to-right, and right-to-left) and subsequent reconstruction in between left and right frames uses the same vector fields. This is in contrast with, for example, pivoting-based reconstruction where a new vector field must be computed for each position of intermediate views.

Finally, we noticed that the smoothness term in spline-based reconstruction partially eliminates blocking artifacts, which are due to compression, and improves the subjective quality.

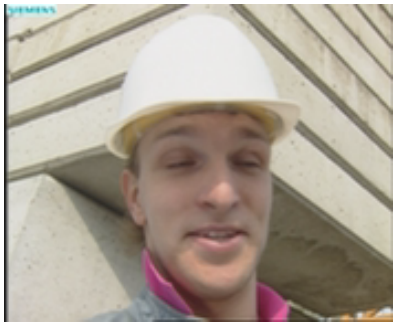
9.2.3 Experimental results

We first present the results on a standard test sequence, Coastguard. We use every third frame of the sequence to reconstruct two frames in between. Since those two frames are available in the video sequence, this allows us to measure the numerical performance of the proposed method. Fig. 9·3.a and b shows the two input frames and Fig. 9·3.c and d shows the two original frames in between. Fig. 9·3.e and f shows the reconstructed frames in between. The difference images between originals and reconstructed views are shown in Fig. 9·3.g and h. Similar results are shown for Foreman sequence in Fig. 9·4. As it can



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 9·3: Coastguard sequence: (a) Frame #1 (b) frame #4 (c) frame #2 (d) frame #3 (e) reconstructed frame #2 (f) reconstructed frame #3 prediction error for (g) frame #2 (33.29dB) (h) frame #3 (33.30dB).



(a)

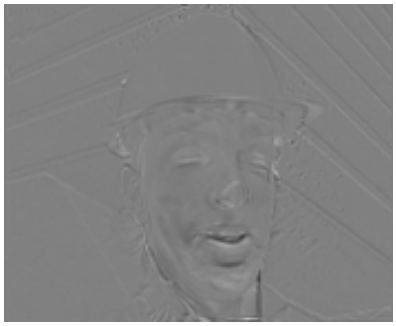
(b)

(c)

(d)

(e)

(f)



(g) 30.3091

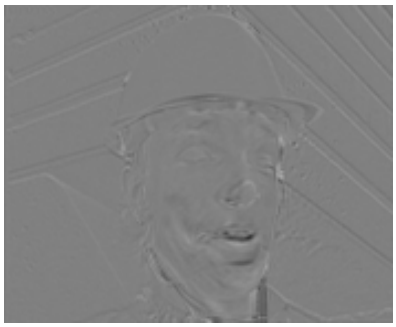


Figure 9·4: Foreman sequence: (a) Frame #1 (b) frame #4 (c) frame #2 (d) frame #3 (e) reconstructed frame #2 (f) reconstructed frame #3 prediction error for (g) frame #2 (34.34dB) (h) frame #3 (34.44dB).

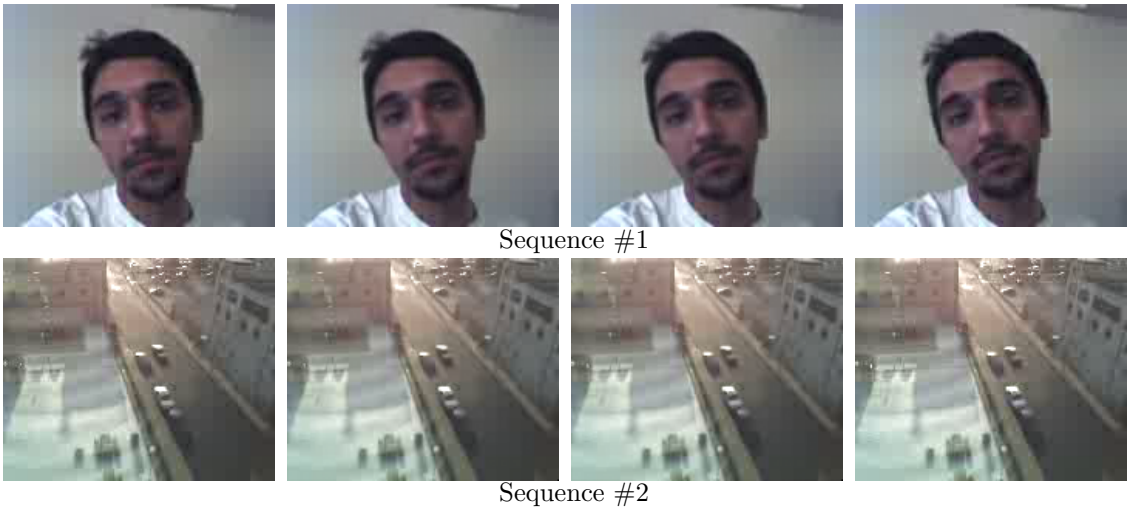


Figure 9·5: First and last columns show input images and middle columns show two of the reconstructions.

be noticed the prediction errors are very small in both experiments. The PSNR values are measured around 33dB. We should not that the distortions around the mouth of the person in Foreman are due to the complex deformable motion of lips. Such a motion cannot be modeled by optical flow.

Next, we captured several video sequences using a Motorola V360 mobile phone. Our measurements reveal that the average frame rate of the videos is around 5.5fps. Considering this rate, we reconstructed five virtual frames between available frames. A subjective evaluation of reconstructed video sequences shows significant improvements; the videos are perceived as smoother. In the first and last column of Fig. 9·5, we show the original frames from video sequence and in the middle two columns we show two of the reconstructed views. The original and enhanced video sequences can be examined at `http://iss.bu.edu/ince/thesis/timeivr.html`.

9.2.4 Conclusions

We presented an application of the proposed view reconstruction algorithms to the enhancement of video sequences captured by mobile phones. The experimental results show the enhanced videos are much smoother and pleasant to the viewer. The method is suc-

cessful because the optical flow method can accurately estimate the motion between video frames.

9.3 Communications: Depth estimation for view synthesis in multiview video coding

In collaboration with Mitsubishi Electric Research Laboratories we worked on coding of multiview video data. The contribution presented in this section was improving block-based disparity estimation to improve compression efficiency (Ince et al., 2007b).

9.3.1 Introduction

Emerging camera arrays (Wilburn et al., 2005) and eye-wear free 3D displays (Dodgson, 2005; Matusik and Pfister, 2004) make 3D TV a feasible product in the future. In an end-to-end 3D system, the transmission and storage of multiple video streams is of concern because of the prohibitive amount of visual data needed. In response to this need, there is currently an MPEG activity on efficient coding of multiview video (JTC1/SC29/WG11, 2005; Vetro et al., 2004).

One of the approaches in multiview coding is to use view synthesis to produce additional references for the view that is being encoded (Kimata and Kitahara, 2004; Martinian et al., 2006b; Martinian et al., 2006a). Consider Fig. 9·6 where we would like to code $I_n(t)$, a frame at time t of camera n. As shown, it is possible to use previous frames, such as $I_n(t-1)$, as references. Also, since the cameras share a common field of view, it is possible to use frames $I_{n-1}(t)$ and $I_{n+1}(t)$ neighboring cameras as references as well. Moreover, by using view synthesis, it is possible to reconstruct a virtual view $V_n(t)$ for camera n using other cameras. Martinian *et al.* (Martinian et al., 2006b; Martinian et al., 2006a) showed that using this synthesized view as an additional reference can introduce notable gains in compression efficiency.

As summarized in Chapter 3, among many methods to synthesize a view, one approach is to compute the depth field of a scene using available cameras and then to use this depth

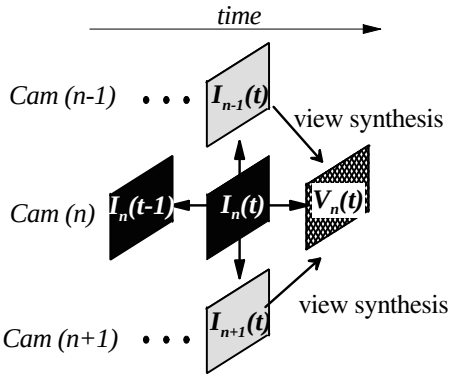


Figure 9·6: Prediction using view synthesis in multiview coding.

map to render a virtual view (Chen and Williams, 1993; Buehler et al., 2001). However, in the case of multiview video coding, one crucial step is the transmission of these depth maps, as they are needed in the decoder. In most cases, the depth of the scene is unavailable and must be extracted. Therefore, when depth maps are computed, the number of bits required to represent them must be considered as well (Alatan and Onural, 1998; Park and Park, 2006). Depth maps for multiview coding should be smooth enough so that they can be coded efficiently, but they should also have enough variations to approximate the scene structure. Considering these needs, in this application, we focus on improving block-based depth estimation to generate smooth and accurate depth maps. We progressively improve the results by introducing a hierarchical scheme, regularization and nonlinear filtering. We also extend the search into color components. These additional steps not only improve the smoothness of depth maps, but also lead to visual improvements in synthesized frames.

First, we will introduce basic depth estimation and then describe improvements to the algorithm. Finally, we will show the efficacy of the resulting depth maps in view synthesis and multiview coding.

9.3.2 Depth estimation for view synthesis

Let A_n, R_n and $\mathbf{t}_n$ denote intrinsic matrix, rotation matrix and translation vector for camera C_n. Given a point $\mathbf{x}_n = [x_n, y_n]$ in image I_n captured by camera C_n and corresponding depth of this point $D(\mathbf{x}_n)$, it is possible to map $\mathbf{x}_n$ onto image I_i from other cameras,

where $i \in \{1 \dots N\}$ and N is the number of cameras. First, the point is projected from two-dimensional image plane into three-dimensional space as follows:

$$\mathbf{X} = R_n \cdot A_n^{-1} \cdot [\mathbf{x}_n \, 1]^T \cdot D(\mathbf{x}_n) + \mathbf{t}_n \tag{9.1}$$

where $\mathbf{X}$ denotes the three-dimensional point. Next, $\mathbf{X}$ can be projected onto desired camera image, for example I_{n-1}, as follows:

$$\mathbf{x}_{n-1} = A_{n-1} \cdot R_{n-1}^{-1} \cdot (\mathbf{X} - \mathbf{t}_{n-1}). \tag{9.2}$$

Combining these two equations we can write $\mathbf{x}_{n-1}$ as a function of $\mathbf{x}_n$ and $D(\mathbf{x}_n)$ within a scaling factor:

$$\mathbf{x}_{n-1}(\mathbf{x}_n, D(\mathbf{x}_n)) = A_{n-1} R_{n-1}^{-1} (R_n A_n^{-1} [\mathbf{x}_n \, 1]^T D(\mathbf{x}_n) + t_n - t_{n-1}) \tag{9.3}$$

Using (9.3), depth estimation seeks to minimize the following prediction error among possible depth values:

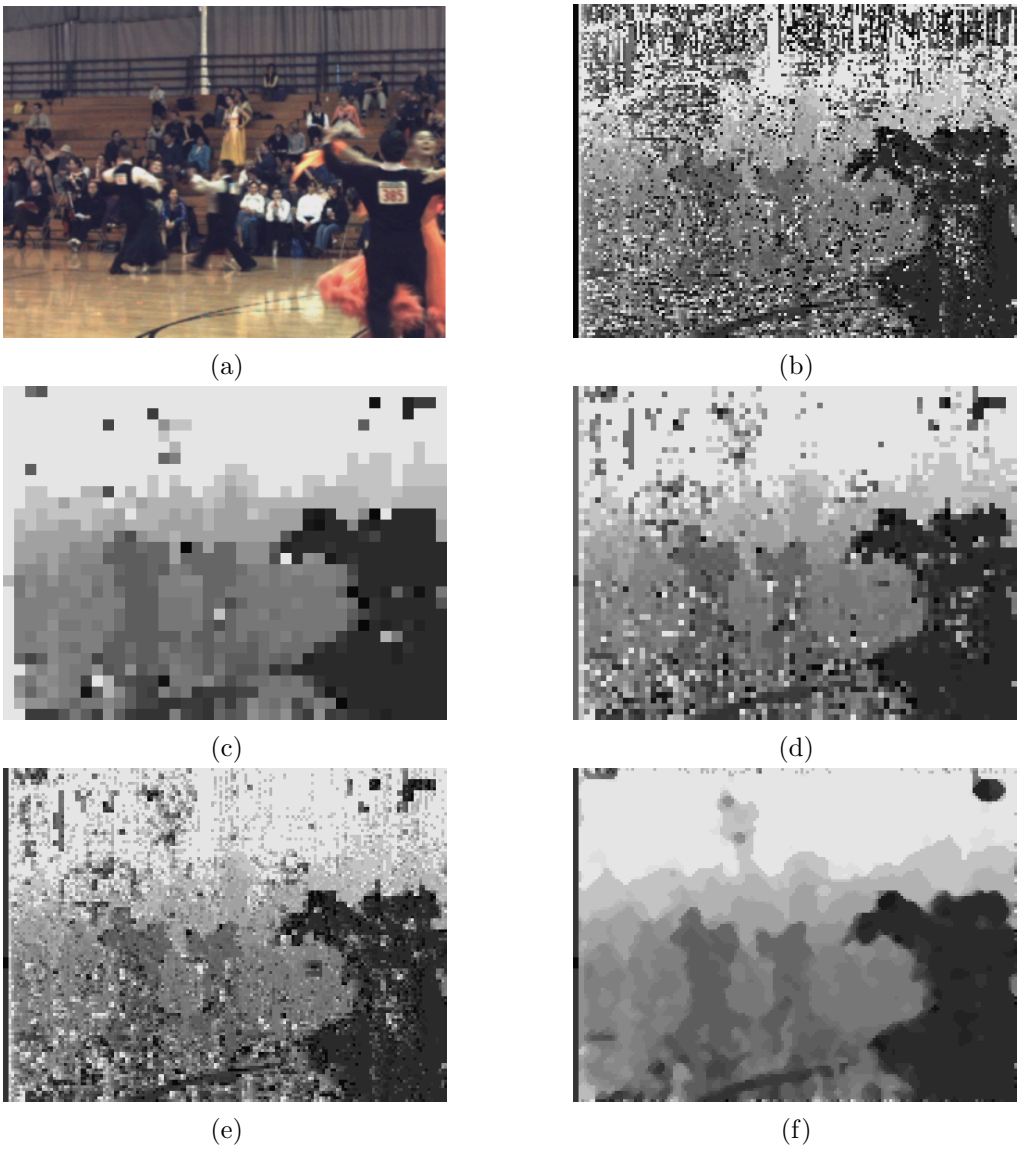
$$P(\mathbf{x}) = \Psi(I_n[\mathbf{x}_n] - I_{n-1}[\mathbf{x}_{n-1}(\mathbf{x}_n, D(\mathbf{x}_n))]) \tag{9.4}$$

where Ψ is an error function, for example quadratic or absolute value function. This can be written as the following minimization:

$$D(\mathbf{x}) = \underset{\{D_i(\mathbf{x})\}}{\arg\min} \, P(\mathbf{x}) \tag{9.5}$$

where $D_i(\mathbf{x}) = D_{min} + iD_{step}$, $i = \{0 \dots (D_{max} - D_{min})/K\}$, and K is the number of possible depth values.

If we consider a block-based model, then the frame I_n is divided into $M \times M$ blocks and prediction error in equation (9.4) is minimized for each block. Since this cost function minimizes the prediction error, it is an excellent choice for compression. However, the resulting depth maps are not suitable for a multiview codec. The problem is that the resulting depth maps are usually very noisy and lack spatial smoothness. One frame from *Ballroom* sequence and its corresponding depth map estimated using 4×4 blocks are shown



(a) (b) (c) (d) (e) (f)

Figure 9·7: Visual comparison of depth maps. (a) View #4, Frame #1 of *Ballroom* sequence. (b) Result of original block-based depth estimation. (c)-(e) Results of hierarchical scheme for each level, $16 \times 16, 8 \times 8, 4 \times 4$ respectively. (f) Final result of improved depth estimation algorithm. Clearly, the improved algorithm generates smoother and more accurate depth maps.

in Fig. 9·7.a and b, respectively. Brighter pixels indicate points that are far away from the camera while darker pixels indicate points that are closer to the camera. Due to the lack of spatial and temporal correlation in these depth maps, conventional compression algorithms fail to achieve a high quality reconstruction while keeping the depth bitrate low. Moreover, the estimated depth values do not accurately represent the scene. It is clear that smoother depth maps are essential to achieve accuracy of depth values and high compression ratios.

9.3.3 Improvements to depth estimation

In this section, we progressively improve the results of the block-based depth estimation algorithm.

Hierarchical Estimation

In the example we show in Fig. 9·7.b, a block size of 4×4 is used to approximate the scene structure. However, carrying only 16 pixels of information, such a block fails to capture local texture essential for finding a good match. Although larger blocks tend to give better matches, they cannot capture local depth variations. Depth maps resulting from large blocks are typically too smooth and blocky. A hierarchical estimation (i.e., coarse to fine) scheme (Grimson, 1985) is a good fit to solve both problems. The algorithm should start from a large block size so that a reasonable, but coarse, depth is estimated and then these values should be used as initial values and refined by smaller block sizes. Specifically, we start with 16×16 blocks, and refine the results using 8×8 and then 4×4 blocks.

Regularization

Regularization, a common tool in inverse problems, introduces *a priori* knowledge to the problem (Karl, 2005). In depth estimation, it can be assumed that neighboring points should have similar depth values because objects are rigid or smooth in real world. Such a constraint is not enforced in equation (9.5), thus yielding noisy depth maps. Therefore, in order to enforce regularization during depth estimation, we introduce a new term that

introduces penalty when a point $\mathbf{x}$ has different depth value than neighboring points:

$$R(\mathbf{x}) = \sum_{k \in \Pi} \Psi(D(\mathbf{x}) - D(\mathbf{x}_k)) \tag{9.6}$$

where Π indicates the neighborhood of the current block and Ψ is an error function. This is a Tikhonov-type regularization, and it is the discrete equivalent of regularization term given in (6.2).

We used second-order neighborhood (eight surrounding neighbors) in the implementation, and absolute value function for Ψ. Combining the prediction error term (9.4) and the new regularization term (9.6), we perform the following minimization:

$$D(\mathbf{x}) = \underset{\{D_i(\mathbf{x})\}}{\arg\min}\, P(\mathbf{x}) + \lambda R(\mathbf{x}) \tag{9.7}$$

where λ is the regularization (smoothness) parameter. Large values of λ result in smoother depth maps. However, it should be noted that increasing λ to very large values yields over-smoothed depth maps which are as unusable as the unregularized ones. Therefore, for best results regularization parameter may need to be adjusted for different sequences. This is a common drawback of regularized methods.

Median filtering

Despite the two previous steps which aim to achieve smooth depth maps, there may still exist outliers in the computed depth map. Median filter is a basic nonlinear filter used to suppress outliers in a data set. We add median filtering to our algorithm as a post-processing step. Once a depth map is computed at each hierarchy level *via* minimization of (9.7), a median filter is applied to eliminate the outliers. We used a fixed window of size 3×3 for median filter mask.

9.3.4 Experimental results and comparison of depth maps

Let us compare the resulting depth maps after each improvement mentioned in the previous section. Results for each step of hierarchy are shown in Fig. 9·7.c-e. Note that after

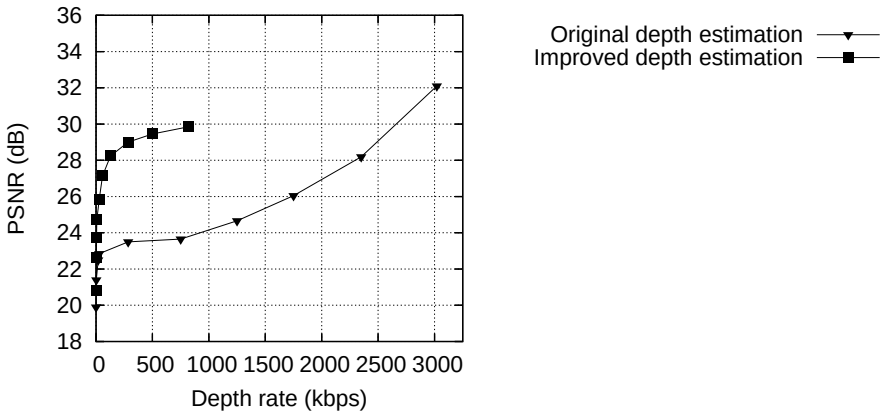


Figure 9·8: Bitrate of the encoded depth field *vs.* synthesized view quality.

successive steps, the depth map provides a better representation of objects in the scene. When compared to the original estimation (Fig 9·7.b), immediate improvements are visible in Fig. 9·7.e, especially in the background. However, this depth map still contains too much variation to be compressed efficiently. The final depth map after regularization and median filtering is shown in Fig. 9·7.f. It is clear that the resulting depth map is much smoother, which should be easier to compress than the noisy depth map in Fig. 9·7.b. We also observe that, subjectively, the depth values are closer to reality. For example, parts of the curtain in the background are detected by the original algorithm as closer points (i.e., darker values in the depth map), which is incorrect. With new algorithm, depth values in those areas are corrected.

As mentioned earlier, smoother depth maps can be compressed more efficiently than noisy depth maps. To verify this claim, we tested synthesized image quality *vs.* bitrate required to encode depth maps using H.264/AVC reference software (JM, 2006) on the *Ballroom* sequence. Results are shown in Fig. 9·8. These results show that the new algorithm outperforms the original algorithm by up to 6dB. The original algorithm is better only at very high (and impractical) bitrates of 3 Mbits/sec.

Finally, we tested synthesized frames generated by the original and improved depth maps in the multiview codec described in (Martinian et al., 2006b). Since bit-rate for depth data was omitted in that study, we focus on the decoded image quality. Compared

Figure 9·9: All colors shown on the left have the same luminance component shown on the right.

to results using depth maps obtained by reference block-based algorithm, we observed approximately the same PSNR using the new depth maps with less than 3% increase in the bit rate. This slight loss in prediction efficiency is expected due to the smoothness constraints imposed by the new algorithm. However, it should be kept in mind that the rate to code the new depth maps will be significantly less.

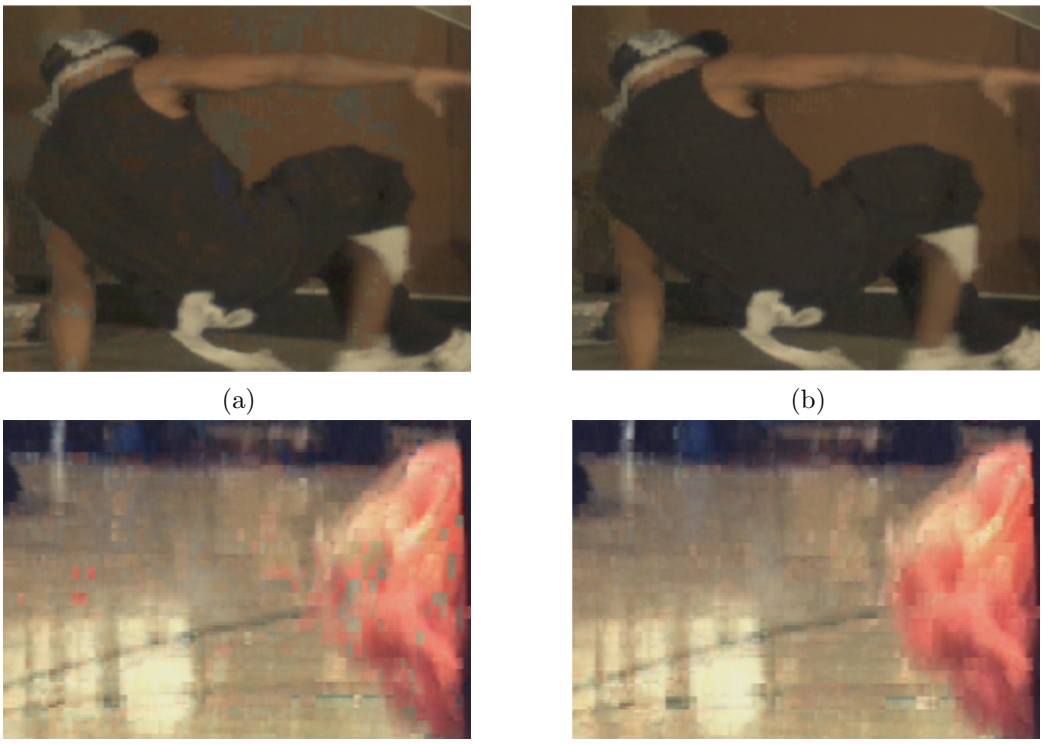
9.3.5 Improvements to visual quality

For the sake of simplicity, usually only one color component, luminance, is used in depth estimation. However, two different textures in an image, especially areas with a smooth color, may have comparable luminance values. For example, in Fig. 9·9, all colors shown have the same luminance value. Due to this, depth estimation may yield incorrect depth values which in turn may result in visual artifacts as shown in Fig. 9·10.a and c (Please refer to electronic version of this dissertation for better quality). Therefore, whether regularized or not, an extension of depth estimation methods to include color components should improve visual quality of the synthesized view. Once minimization in equation (9.7) is carried out on luminance and chrominance components jointly, such artifacts are significantly reduced as shown in Fig. 9·10.b and d.

9.3.6 Conclusions

In this application, we considered the estimation of smooth and reliable depth maps for view-synthesis-based multiview coding. By adding several improvements, we showed that such depth maps improve both compression efficiency and visual quality.

Further improvements in the depth estimation might be achieved by using variable block sizes instead of fixed sizes (Mancini and Konrad, 1998). Synthesis correction vectors



(a) (b)

(c) (d)

Figure 9·10: Synthesis results (a,c) without and (b,d) with using YUV search. (*Breakdancers* is property of Microsoft.)

(Martinian et al., 2006a) can improve the results as well.

Currently, the algorithm uses a fixed number of possible depth values and it linearly samples the depth range. Obviously, the number of possible depth values directly affects the synthesized image quality and depth maps. Therefore, a mechanism to adjust the depth range depending on available bandwidth may be considered. Moreover, linearly sampling the depth may not be always effective to approximate a scene. For example, objects closer to the camera will have more visible depth variations than objects far away, but possible depth values may not cover all structural details of this closer object and this may lead to artifacts in synthesized view. Visually, artifacts on objects closer to camera will have more perceptual impact. Thus, nonlinear quantization of depth with emphasis on small depth values should be considered.

9.4 Space exploration: Recovery of 3D images of Mars (or helping NASA find little green Martians)

This application demonstrates our methods applied to a real-world problem faced by NASA (Ince and Konrad, 2005b). We use the proposed occlusion-aware optical flow estimation and subsequently spline-based view reconstruction algorithm to reconstruct full-color 3D images of Mars on automultiscopic displays.

9.4.1 Introduction

The recent mission of two NASA rovers, "Spirit" and "Opportunity", to Mars has provided a lot of information about the planet. In order to remotely explore the surface of Mars, "Spirit" and "Opportunity" have been equipped with a cutting-edge stereo camera called *PanCam* (Bell III and *et al.*, 2003b). *PanCam* has been designed to take multi-spectral stereo pictures in order to help scientists in their quest for discovery of life on Mars. While commercial cameras usually capture three spectral bands, namely red (R), green (G) and blue (B), *PanCam* is sensitive to more bands since this information is valuable to geologists. This is achieved by means of a filter wheel in front of each camera lens (Fig. 9·11(a)). Slightly different filters are used on wheels of both cameras; while the left camera is equipped with red, green and blue filters, among others, the right camera does not have a green filter on its color wheel. Therefore, since the G component of the right image is missing, currently it is not possible to view a 3D image of Mars surface in color. In this application, we address the issue of recovery of this missing component and reconstruction of intermediate views to be displayed on a automultiscopic 3D display.

9.4.2 Why is green component missing?

The *PanCam* camera captures different colors by using the filters on a rotating wheel placed in front of the left and right lenses (Fig. 9·11(a)). The table in Fig. 9·11(b) lists the center wavelength λ_c of each filter and its bandwidth $\Delta\lambda$. The filters have been designed for multispectral sky imaging, direct Sun imaging and also for geologic and mineralogic studies

(a)

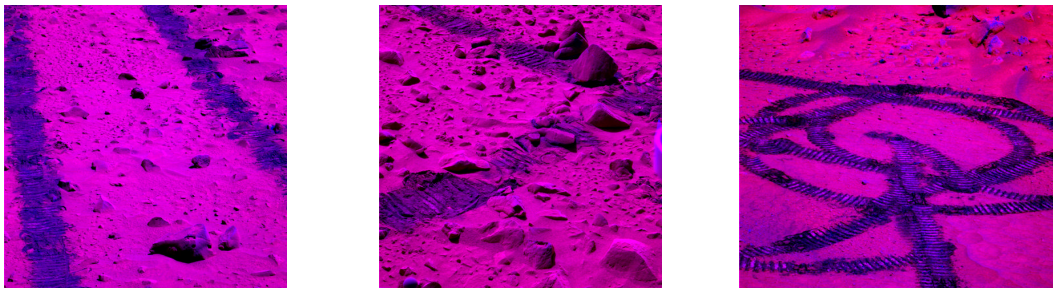
Left camera: λ_c ($\Delta\lambda$)		Right camera: λ_c ($\Delta\lambda$)	
L1	739nm (338nm)	R1	436nm (37nm)
L2	753nm (20nm)	R2	754nm (20nm)
L3	673nm (16nm)	R3	803nm (20nm)
L4	601nm (17nm)	R4	864nm (17nm)
L5	535nm (20nm)	R5	904nm (26nm)
L6	482nm (30nm)	R6	934nm (25nm)
L7	432nm (32nm)	R7	1009nm (38nm)
L8	440nm (20nm)	R8	880nm (20nm)

(b)

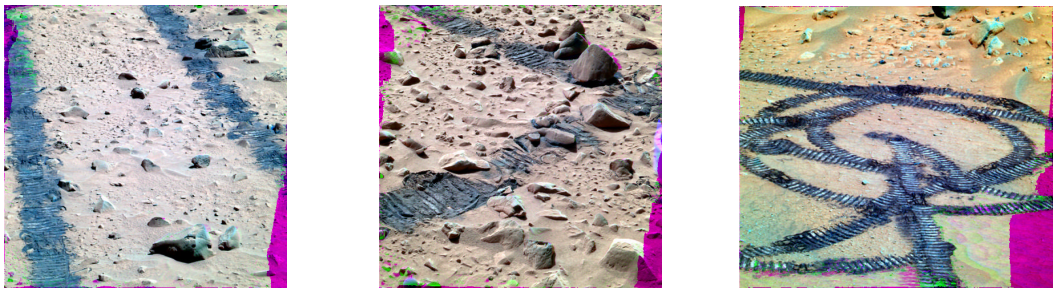
Figure 9·11: Spectral characteristics of the *PanCam*: (a) photo of *Pancam* and its rotating filter wheels, and (b) central wavelengths and bandwidths of individual filters (from `http://marsrover.nasa.gov`).

of Mars surface (Bell III and *et al.*, 2003b), rather than for surface color reproduction. In order to reproduce color from the Mars surface, information from three parts of the spectrum is needed, for example close to the the 1931 CIE primaries (red: 700nm, green: 546.1nm, blue: 435.8nm). Excluding the wideband filter L1, that covers close to half of the visible spectrum, a good choice of filters to reproduce left-image color is: L2 for red, L5 for green and L7 for blue. Similarly, for the right image a good choice is: R2 for red and R1 for blue. However, there are no filters in the right image close to the CIE green primary . Since the green component is missing, it is not possible to create a full-color stereo pair of the Mars surface. Therefore, the goal of our research is to first develop a method capable of recovery of the missing green component of the right image using all three components of the left image, and the available components of the right image. Then, we would like to use the reconstructed stereo images to reconstruct intermediate views to displayed on a multiview 3D display.

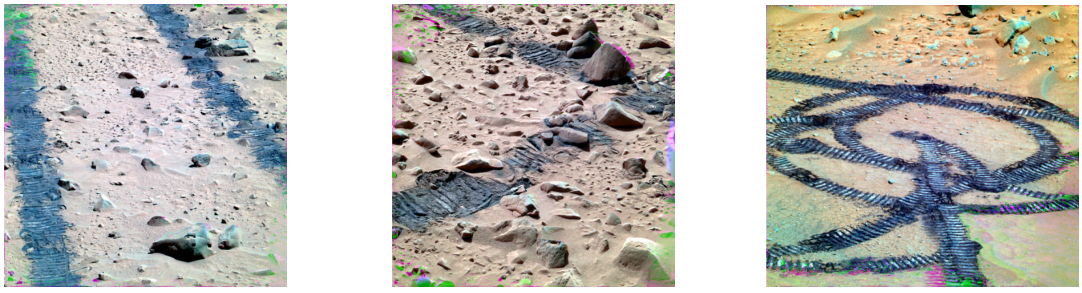
Left images with all available color components



Right images with G components padded with zeros (i.e., unavailable)



Right images after disparity-compensated prediction of G component using occlusion-aware optical flow estimation



Final reconstructed right images after filling missing areas using image-driven image inpainting

Figure 9·12: Reconstruction of Mars images. Rows from top to bottom show left images; right images with unavailable G components; right images after disparity-compensated prediction of the G component, and finally right images after filling in missing areas.

9.4.3 Recovery of a missing color component in stereo images

As we have seen in Chapter 2 when a 3D scene is acquired by a pair of color-sensitive cameras, each 3D point is projected onto the sensor plane of each camera creating the so-called homologous pair of points. These points inherit all photometric properties of the original 3D scene point. Depending on the color space each camera uses, the photometric information at camera's output may be luminance and chromaticities, or red, green and blue tristimulus values. The projection geometry, however, is independent of the photometric properties of the 3D point, and thus all components (whether luminance and chromaticities, or red, green and blue channels) share the same disparity.

Thus, if we recover a disparity field from only some color channels of a stereo pair, we know that this disparity field also applies to the other channels of this stereo pair. Therefore, the goal is to compute a disparity field using the available R and B components, and then derive the G component of the right image from the G component of the left image. The main step of this reconstruction algorithm is disparity estimation. We will use the proposed occlusion-aware optical flow to estimate the disparity. Then, the recovery of G component can be performed by means of disparity-compensated prediction.

Once the disparity field $\{\mathbf{d}\}$ has been computed for all pixels of the right image I_R, a precise correspondence between features in the left and right images for all color components is known. Thus, value of color (tristimulus value) $i \in \{R, G, B\}$ at location $\mathbf{x}$ in the right image, namely $I_{R,i}(\mathbf{x})$, corresponds to the value of the same color at location $\mathbf{x} + \mathbf{d}(\mathbf{x})$ in the left image, i.e., $I_{L,i}(\mathbf{x}+\mathbf{d}(\mathbf{x}))$. Although these two values may not be identical, they, in general, should be very similar, except for areas where disparity estimation may fail (e.g., image boundaries). Thus, $I_{R,i}(\mathbf{x})$, can be accurately predicted from $I_{l,i}(\mathbf{x} + \mathbf{d}(\mathbf{x}))$.

In particular, the green component of the right image can be recovered by the above mechanism, often called disparity-compensated prediction, as follows:

$$I_{R,G}(\mathbf{x}) = \widetilde{I}_{L,G}(\mathbf{x} + \mathbf{d}(\mathbf{x})), \quad \forall \mathbf{x} \in \Lambda_R \tag{9.8}$$

where Λ_R is the domain of I_R, $I_{R,G}$ is a predicted value of the green component, and $\widetilde{I}$ denotes interpolation of intensity I since the position $\mathbf{x} + \mathbf{d}(\mathbf{x})$ need not belong to the sampling grid of the left image.

9.4.4 Experimental results

We applied the proposed algorithm to a number of stereo pairs from the Mars mission[1]. In each pair, all components of the left image, and the R and B components of the right image were available, and we have reconstructed the missing G component. The Mars images are not captured with a parallel camera setup, however optical flow estimation successfully captured vertical disparities.

The original color left images are shown in the first row of Fig. 9·12. The second row shows the right images without the G component. In this case we simply filled the G component with zeros. The third row shows right images predicted using equation (9.8). The structures are very well matched and visually pleasing. However, there are gross errors near the boundaries of the image. In close inspection it is obvious that those areas are visible only in the right image, not in the left image. Since equation (9.8) cannot fill these areas in, they have different color than the surroundings.

In order to solve this problem, we utilize the image-driven disparity inpainting proposed in Chapter 5. Previously, we inpainted missing areas in a disparity map by using gradient of the underlying image. Now, we inpaint the missing areas in a color component by using the gradient of another color component. The idea is that edge information is independent of color components and can be used in guiding the diffusion process. The results after inpainting are shown in the last row of Fig. 9·12. As it can be noticed, the images are of very high quality. Although there are still a few discolorations around boundaries of the image, this is expected, because diffusion alone is not sufficient to create texture in those areas.

[1]Mars images can be downloaded from NASA Jet Propulsion Laboratory at `http://marsrovers.jpl.nasa.gov/gallery/all/`

After the reconstruction of stereo images, we created seven additional views in between and displayed them on an automultiscopic display (SynthaGram SG202). The resulting 3D experience was comfortable and colors were well matched.

9.4.5 Conclusions

In this application, we applied the optical flow method proposed in Chapter 6 to the reconstruction of a missing green component of right image in a stereo pair. Then, we used these stereo pairs to generate 3D images of Mars on an automultiscopic 3D display.

This application once again proves the versatility of our methods. Without any calibration or additional information, we were able recover a stereo pair and intermediate views. It should also be noted that the input images were not captured by parallel cameras, and therefore included vertical disparities. Yet, our algorithms successfully handled this additional problem.

Chapter 10

Conclusions and future work

This chapter summarizes the contributions of this dissertation, outlines the potential applications and proposes ideas for future work.

The main inspiration for this work was the emergence of multiview eyewear-free 3D displays, or shortly, automultiscopic displays. Although these displays promise a revolutionary way of 3D visualization, content to be reproduced is quite scarce; these displays require several views of a scene that are often unavailable. This is in contrast to stereo displays that require two views only. Considering this problem, we focused on the reconstruction of intermediate (virtual) images from stereo images. Intermediate view reconstruction has other applications in 3D systems such as enhancement of viewer comfort or transmission of multiview video, or even in monoscopic video sequences for frame-rate conversion.

In this work, we focused on three main issues in intermediate view reconstruction: proposing an alternative to the backward-projection methods, improving methods based on backward-projection, and finally, handling of occlusion areas in view reconstruction.

As we mentioned earlier, the prior work often focused on pivoting-based or backward-projection methods. These methods reconstruct individual pixels of an intermediate view by pivoting on the sampling grid of intermediate view and backward-projecting this point onto input images. These methods have two deficiencies. The first issue is related to the way disparity of the intermediate view is computed. Since the intermediate view is not known during disparity estimation, disparity is estimated for an *unknown* image and then this disparity is used to reconstruct this unknown image. Experimental results show that the disparity estimated using this approach is usually not of high accuracy. More specifically, shapes of objects are not preserved mainly because of the absence of

an underlying image; had the underlying image been available, disparity estimation could have utilized the underlying image gradient to regularize disparities and, therefore, generate disparity maps with sharp discontinuities (at object boundaries). Another deficiency of this approach is that a new disparity map must be computed for each intermediate view, thus imposing additional computational burden.

In our work, we proposed a forward-projection method to complement and solve deficiencies of backward projection. The main difference in this case is that the disparity is estimated for the *known* images, e.g., left and right images in stereo. Then, intensities of input images are forward-compensated by using the estimated disparity field in order to reconstruct the intermediate view. The main advantage in this case is that more accurate disparity values can be estimated, because disparity estimation can utilize the underlying image gradient during regularization. Forward-compensation methods were proposed in the past, but the final reconstruction stage was less sophisticated than the spline-based method that we used. Our forward-compensation method has a computational advantage as well. A single disparity field is sufficient to reconstruct any intermediate view, i.e., there is no need to estimate a new disparity field for each intermediate view.

The second main focus of the dissertation were occlusions that occur when some image areas are visible only in one of the views. We intended to introduce occlusion awareness into individual steps of view reconstruction in order to improve reconstruction quality in occlusion areas, especially since reconstruction in these areas are usually overlooked by other methods. Another reason for our focus on occlusions is that they occur in monoscopic video sequences as well and, therefore, our methods would be applicable to other video processing problems as well, thus having a broader impact.

Our main contribution was to address the interrelation between occlusions and disparities: occlusions occur due to scene structure (depth), that is related to disparities, while accurate recovery of disparities requires knowing location of occlusions. Traditionally occlusion areas and disparity fields were computed separately since it was unclear how to create a feedback between them. Usually, one was estimated and the result was used in

the computation of other and vice versa. Instead of such a step-by-step approach, we proposed a method that jointly estimates disparities, implicitly estimates occlusion areas and assigns plausible disparity values to occlusion areas. The proposed method is the first joint occlusion-aware optical flow formulation to the best of our knowledge and is equally applicable to stereo and video. In the process, we also developed a simple, yet effective, method for the estimation of occlusions based on the diverging nature of disparity fields around newly-exposed areas. An interesting feature of this method is the fact that it can predict occlusion areas between the frames from which the disparity was estimated.

After this high-level discussion of contributions of the thesis, we would like to elaborate on the methods proposed in the dissertation in more detail in the next section.

10.1 Detailed discussion of technical contributions

After analyzing the view reconstruction problem, we concluded that there are five essential steps of view reconstruction: estimation of disparity, estimation of occlusions, handling of occlusion areas, identifying which points are visible in intermediate view and finally estimation of texture. In this dissertation, we addressed all of these problems and proposed novel methods. One of the most important properties of these methods is that they are not limited to stereo, but are directly applicable to video sequences with respect to time axis.

First, for the estimation of texture, we proposed a spline-based view reconstruction algorithm by extending a method for irregular-to-regular image interpolation. The proposed method uses intensity values of left and right images to reconstruct a new intermediate image. Our experimental results show that this method is superior to a reference pivoting-based method. The main advantage of using splines and therefore forward-compensation of intensities is that it permits to selectively use input images. This is advantageous because, if a point is not visible (i.e., is occluded) in the intermediate image, it should not be used by the reconstruction method at all (it should not be forward-compensated). Therefore, this method is suitable for handling occlusion areas.

Next, we focused on how to estimate which points are visible/occluded. In order to estimate occlusion areas, we first proposed a simple, yet very effective method. This method relies on a simple fact: when a disparity field is computed between two images, pixels in the target frame that did not exist in the reference frame (i.e., newly-exposed pixels) have no relationship with the reference frame and, as such, cannot be pointed to by forward disparity vectors. Thus, when pixels of the reference frame are forward disparity-compensated onto target frame, these areas are void of disparity-compensated projections. Such areas can be detected relatively easily and are equivalent to occlusion areas when target and reference frames are interchanged. Our experimental results show that the method is very reliable and more robust to image noise than other methods. One of the most important properties of this occlusion estimation method is that it allows to estimate areas that will be occluded/exposed in the intermediate image as well. We need such a detection method, because occlusion areas between the left and the right images are not fully occluded in the intermediate image (please see Fig. 3·1 for an example). Therefore the proposed method can be used to estimate visibility of the points in the input, as well as, in the intermediate images.

Unfortunately, even if we can estimate occlusion areas, disparity in these areas is still unknown, because it is not possible to match such areas between images. Yet, we do need to know the disparity in occlusion areas, because without that information, it is not possible to reconstruct partially visible areas in the intermediate view. Considering the recovery of disparities in occlusion areas, we proposed image-driven disparity inpainting. This method diffuses the available disparities into occlusion areas by utilizing the underlying image gradient. Since the diffusion is guided by image gradient, the resulting depth maps exhibit sharp object boundaries; a very valuable characteristic. Our experimental results show that image-driven disparity inpainting is more reliable than other methods such as depth constancy or standard image inpainting work of Bertalmio *et al.*(Bertalmio et al., 2000).

Although the estimation of disparities and occlusions, and the handling of occlusions can be treated as separate steps, in fact, they are closely interrelated. Specifically, the knowl-

edge of occlusion areas can be used to estimate more reliable disparities which should lead to more reliable occlusion estimates. It is clear that we should allow interaction between disparities and occlusions. Considering this, we proposed an occlusion-aware disparity (optical-flow) estimation algorithm that facilities this interaction.

As we mentioned earlier, the main property of occlusion areas is that they are visible only in one of the images. Therefore, it is not possible to match a point in an occlusion area to a point in an other image. However, since disparity estimation methods are unaware of occlusion areas at the beginning of estimation, these areas are forced to be matched with points in an other image. Therefore, they are assigned incorrect disparity values. Even worse, due to spatial regularization these areas adversely affect neighboring visible areas as well. Therefore, we proposed an occlusion-aware formulation where the prediction term, which matches intensities of the points, is disabled if a point is detected as an occlusion point. The decision if a point is an occlusion point is made by measuring forward and backward compatibility of the disparity fields from left to right and from right to left. The formulation is designed so that if the prediction term is disabled, then image-driven inpainting dominates and fills in disparity of occlusion areas by using available neighboring of visible points. Experimental results of such occlusion-aware optical flow estimation show significant improvements over other state-of-the-art optical flow methods. The method is also shown to be robust to image noise. To our best knowledge, this is the first optical method that jointly estimates occlusions and optical flow. This method also addresses three problems of view reconstruction simultaneously, namely, estimation of disparity, estimation of occlusions and handling of occlusions.

Finally, after proposing solutions for each sub-problem of view reconstruction, we combined all these methods to achieve spline-based occlusion-aware intermediate view reconstruction. Results on both synthetic and real images show high-quality results.

As a final contribution, we focused on extending pivoting-based reconstruction to handle occlusions. We decided to use multiple images instead of stereo pairs to solve the occlusion problem. Actually, there are two main issues hampering the pivoting-based methods. The

first one is that no edge-preserving regularization can be applied because the underlying image is unavailable. The second issue is that it is unclear how to handle occlusions. We first proposed to use a coarse image obtained by pivoting-based method to achieve edge-preserving regularization. This significantly improved the accuracy of estimated disparities in visible areas, but did not offer any advantage in occlusion areas. Therefore, we proposed a variational approach to estimate the disparity by using multiple images. The variational method is adaptive to occlusions in that it uses different pairs of input images to estimate the disparity; the point of interest is always visible in both images. Our experimental results show that this new multi-view pivoting-based method outperforms pivoting-based methods that use two images.

In the final part of the thesis, we showed several applications of the methods proposed in this dissertation.

First, we used the proposed view reconstruction algorithms to enhance the videos captured by mobile phones. Such videos usually have very low frame rates. We interpolated the missing frames in order to reconstruct 30 frames per second videos that look more natural to the viewer. In another application, we utilized the proposed view reconstruction algorithm in a neuromuscular training system. We equipped the system with 3D displays and presented to a patient 3D images of his/her home environment. The 3D images were generated from stereo pairs captured by medical personnel by using our view reconstruction algorithms. The system is soon to be used by bedridden patients in order to improve their motor functions. In another application, we improved disparity estimation stage of a multiview video codec, which utilizes view reconstruction, for better performance. In the final application, we solved a real-world problem faced by NASA in the Mars mission. By utilizing the occlusion-aware optical flow algorithm, we recovered a missing color component of a stereo pair. Then, we used proposed view reconstruction algorithm to create additional views of the scene to display Mars images on an automultiscopic monitor. All these applications demonstrate the versatility and broad applications of the methods developed.

Overall the contributions of this dissertation can be summarized as follows:

- Analysis of a widely used pivoting-based approach to intermediate view reconstruction
- Development of a robust and relatively accurate, yet simple, occlusion detection method
- Analysis of interrelation between occlusions and disparities and development of a method that simultaneously estimates disparities and occlusion areas and extrapolates the disparities of visible areas into occlusion areas
- Development of a selective process combined with spline approximation for view reconstruction
- Development of a variational approach for view reconstruction that uses multiple images and adaptively reconstructs intermediate images
- Application of view reconstruction algorithms on video sequences for enhancement purposes

10.2 Future work

This section briefly describes a few directions that could extend our work.

10.2.1 Improving occlusion-aware optical flow

The occlusion-aware optical flow algorithm uses an underlying image gradient for diffusion. The formulation forces disparity discontinuities to coincide with high-magnitude image gradients assuming that object boundaries demonstrate such high-magnitude gradients. However, this is not always valid; texture of an object may have high gradient values as well. Therefore, in rare cases, diffusion is incorrectly inhibited by this false edge information. A segmentation algorithm that successfully estimates object boundaries can improve the results.

In a related problem, if an occlusion area contains significant texture, the extrapolation of disparities into occlusion areas, i.e., image driven disparity inpainting, is inhibited. A method which decomposes the image into texture and structure, such as in the paper by Vese and Osher (Vese and Osher, 2003), may solve this problem.

10.2.2 Real-time implementation of proposed methods

The implementation of spline-based reconstruction is not fast enough to achieve a real-time reconstruction today. Currently Gauss-Seidel iterations are used to minimize the cost function; alternative methods such as conjugate gradient may improve the execution speed.

Similarly, the implementation of occlusion-aware optical flow algorithm can be improved. In our implementation, we explicitly discretized the Euler-Lagrange equations, because explicit discretization is straight-forward and simple to implement. However, despite its simplicity, explicit discretization requires a small time step to ensure stability of the solution. It is possible to use semi-implicit or implicit discretization, which will remove the restriction on time step and may allow faster implementation. However, implicit discretization requires solving a system of linear equations, which is an additional burden for the implementation.

Finally, all the methods were implemented partially in Matlab, to take advantage of some of the built-in functions, and partially in C, for its computational efficiency. The parts that are implemented in Matlab are usually slow. Those parts can be ported to C for a better performance.

10.2.3 Embedding spline-based reconstruction into variational formulation

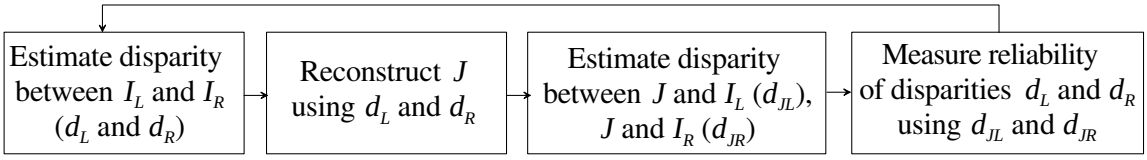


Figure 10·1: A feedback loop for disparities can be constructed if spline-based reconstruction is embedded into disparity estimation.

Currently, the proposed spline-based reconstruction is a separate step in the overall view reconstruction method. It is worth investigating the possibility of embedding this reconstruction into the disparity estimation stage. This will also allow the reconstructed image to interact with the estimated disparity. Specifically, the reconstructed image can be used to measure how reliable the disparity values are (Fig. 10·1). Such a feedback in the formulation may lead to improved performance.

10.2.4 Mathematical representation of the proposed occlusion estimation method

The proposed occlusion detection algorithm in Section 5.3 can be represented by a mathematical model and used in occlusion-aware optical flow estimation. Currently, the occlusion detection method assumes that the image is composed of discrete points, therefore using such a model in a variational framework is not possible. However, if one can represent the estimation method in continuous domain, then it would be possible to utilize the method in occlusion-aware optical flow.

The continuous formulation could be as follows. To explain the idea better, let us first focus on 1D case. Extension to 2D will be shown later.

Let $\Omega(x)$ be a function that measures the density of the points and be defined as follows:

$$\Omega(x) = \sum_{x'=0}^{N_x} \Pi(x - x'), \tag{10.1}$$

where x' is a grid point, N_x is the number of points and $\Pi(x)$ is the rectangular function shown in Fig. 10·2.a. Since this function is not differentiable, one can use the following continuous approximation:

$$\Pi(x) = \frac{1}{2}\left[\tanh(K\,(x + w)) - \tanh(K\,(x - w))\right]. \tag{10.2}$$

As the constant $K \to \infty$, this function approaches the rectangular function with w determining the width of the rectangle. For example, when $K = 10^3$ and $w = 1/2$ the function approximates ideal rectangular function very well as shown in Fig. 10·2.b.

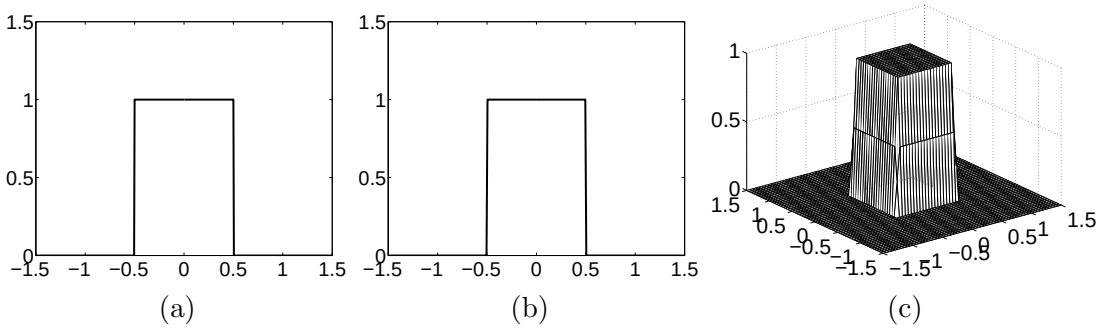


Figure 10·2: (a) Perfect rectangular function; plot of (b) $\Pi(x)$ in equation (10.2) (c) $\Pi(\mathbf{x})$ in equation (10.4). $K = 10^3$ and $w = 1/2$ in (b) and (c).

It is easy to notice that with this definition of $\Pi(x)$, $\Omega(x)$ will be constant i.e., $\Omega(x) = 1$, $\forall x$, because Ω will be composed of the shifted replicas of rectangular function. Now consider that we estimated a disparity field $d(x)$. The disparity compensated density function $\Omega(x+d(x))$ would no longer be a constant; there will be areas where $\Omega(x+d(x)) > 1$ because occluded points will fall in a neighborhood. On the contrary, there will be areas where $\Omega(x + d(x)) = 0$ which are the exposed areas.

Therefore, by thresholding the value of $\Omega(x + d(x))$, it is possible to conclude whether x is an exposed area or not. Considering this idea, one can replace $D(\epsilon(\mathbf{x}))$ in (6.9), with this new detection formula.

Finally, let us show two dimensional representations of $\Omega(x)$ and $\Pi(x)$. We can write $\Omega(\mathbf{x})$ as follows:

$$\Omega(\mathbf{x}) = \sum_{x'=0}^{N_x} \sum_{y'=0}^{N_y} \Pi(x - x', y - y'), \tag{10.3}$$

where (x', y') is a grid point, N_x and N_y are the number of columns and rows respectively and $\Pi(\mathbf{x})$ is defined as a separable function as follows (Fig. 10·2.c):

$$\begin{aligned} \Pi(\mathbf{x}) = \quad & \frac{1}{2}\left[\tanh(K\,(x + w)) - \tanh(K\,(x - w))\right] \times \\ & \frac{1}{2}\left[\tanh(K\,(y + w)) - \tanh(K\,(y - w))\right]. \end{aligned} \tag{10.4}$$

10.2.5 Extension of view reconstruction method to large-baseline cameras

The methods proposed in this dissertation are applicable to cameras with a small baseline. Especially, the proposed optical flow algorithm will have limited performance if the cameras are positioned far from each other. Another future direction could be extending the proposed algorithms to large-baseline cameras.

The main problem encountered in images captured by large-baseline cameras is that the common field-of-view is significantly smaller. Therefore, points near image boundaries are usually invisible in the other image. This is, in fact, another form of occlusion. However, these occlusions are severe; as much as half of the image points may be occluded between images.

Let us suggest some future directions to solve this problem. First of all, since occlusion areas are large, there must be several views available, so that all points are visible at least in two images.

Moreover, another input information that would be required is the camera calibration data. If camera locations in 3D world are known, then it may be possible to use geometric relations, as given in Section 9.3, to estimate if a point is visible in other images.

It is also possible to embed camera calibration data into optical flow formulation as proposed by Alvarez *et al.* (Alvarez et al., 2002b). However, formulation of Alvarez *et al.* does not estimate the visibility of points in input images, therefore it will have problems around boundaries as well. As we proposed in Chapter 6, a term that estimates visibility of points by using geometric relations and camera calibration information can be embedded into the formulation. Such a method may successfully label the points that are not visible between images. The formulation can further be extended to use other image pairs for matching, as proposed in Chapter 8.

Appendix A

Camera geometry

A.1 Pinhole camera

Consider point $\mathbf{X} = (X, Y, Z)$ in 3D space in Fig 2·2.a and its projection at $\mathbf{x} = (x, y)$ onto image plane R. Using similarity of triangles $(C, \mathbf{X}, \mathbf{Y})$ and $(C, \mathbf{x}, \mathbf{y})$, following relations can be written

$$\frac{Z - f}{f} = -\frac{X}{x} = -\frac{Y}{y}. \tag{A.1}$$

Rearranging this equation x and y can be written as follows:

$$x = -f\frac{X}{Z - f}, \quad y = -f\frac{Y}{Z - f}. \tag{A.2}$$

A.2 Parallel cameras

Let us study projection of point $\mathbf{X} = (X, Y, Z)$ in 3D space onto left and right cameras, at $\mathbf{x}_L$ and $\mathbf{x}_R$ respectively (Fig. A·1) and derive the disparity equations. Focal length of cameras is denoted by f and the baseline distance by b.

The projection onto the x axis of left camera is shown in Fig. A.2. By using the similarity of triangles $(\mathbf{x}_L, T, \mathbf{X})$ and $(C_L, T', \mathbf{X})$, the following can be written

$$\frac{b/2 - X}{b/2 - X + x_L} = \frac{Z - f}{Z}. \tag{A.3}$$

Rearranging this equation $\mathbf{x}_L$ is found as follows:

$$x_L = f\frac{X - b/2}{f - Z}. \tag{A.4}$$

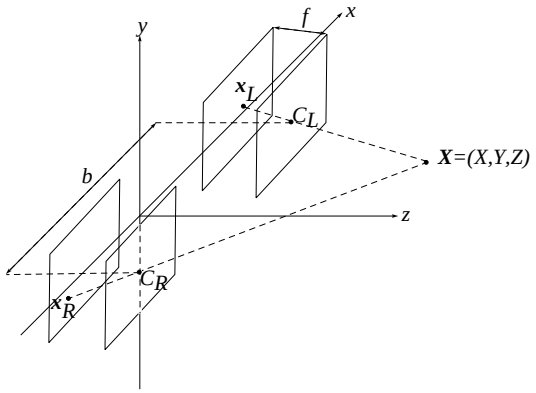


Figure A·1: Parallel cameras.

Similarly, x_R can be computed as follows:

$$x_R = f\frac{X + b/2}{f - Z}. \tag{A.5}$$

The projection onto the y axis of left camera is shown in Fig. A.2. By using the similarity of triangles $(\mathbf{y}_L, T, C_L)$ and $(\mathbf{X}, T', C_L)$, it can be written as:

$$\frac{f}{Z - f} = -\frac{y_L}{Y}. \tag{A.6}$$

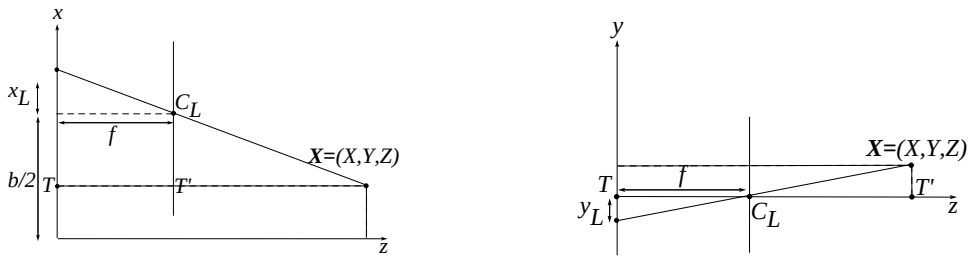
Rearranging this equation y_L, and similarly y_R, are found as follows:

$$y_L = f\frac{Y}{f - Z}, \quad y_R = f\frac{Y}{f - Z}. \tag{A.7}$$

Using these values the disparity can be computed as

$$\mathbf{d}_{LR} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} - \begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} f\frac{X-b/2}{f-Z} \\ f\frac{y}{f-Z} \end{bmatrix} - \begin{bmatrix} f\frac{X+b/2}{f-Z} \\ f\frac{y}{f-Z} \end{bmatrix} = \begin{bmatrix} -f\frac{b}{f-Z} \\ 0 \end{bmatrix}. \tag{A.8}$$

Note that vertical component of disparity is equal to zero. Finally, using similarity of triangles in $(\mathbf{x}_L, \mathbf{x}_R, \mathbf{X})$ and $(C_L, C_R, \mathbf{X})$ in Fig. A·1, the depth of the point is defined as



(a) (b)

Figure A·2: Projection onto (a) x (b) y axis of left camera.

follows:

$$\frac{Z-f}{Z} = \frac{b}{b+d} \tag{A.9}$$

$$Z = f\frac{b+d}{d}. \tag{A.10}$$

Another concept in camera geometry is the epipolar plane which is the plane defined by C_L, C_R and $\mathbf{X}$ (Fig. A·1). The intersection of epipolar plane with an image plane forms a straight line, called the epipolar line. Epipolar line is the projection of a ray through the optical center and image point of one camera, onto the other camera. For example, projection of ray passing through x_R, C_R and $\mathbf{X}$, creates the epipolar line in the left camera.

The knowledge of epipolar geometry is important. Given an image point in one camera, the homologous point in the other image must lie on the epipolar line in the other image. Therefore, given camera geometry information, the search for homologous points can be reduced to a 1D search problem rather than a 2D one. In the special case of parallel cameras, epipolar lines coincide with image scan lines.

A.3 Toed-in (converging) cameras

In toed-in setup, the cameras are rotated to each other by a small angle of θ as shown in Fig. 2·2.b. Unlike parallel cameras, the optical axes intersect at a physical point instead of

infinity. The depth of convergence point can be computed as follows:

$$Z_{conv} = \frac{b}{2}\tan(\theta). \tag{A.11}$$

The projection of 3D points onto cameras is now more complicated since the camera planes are rotated around y axis by θ (left camera) and $-\theta$ (right camera) degrees. Instead of deriving the disparity equations using similarity of triangles, let us use available results of projective geometry (Faugeras, 1993; Franich, 1996; Hartley and Zisserman, 2004). Rotation matrix around y axis with θ degrees is defined as follows:

$$R_y = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \tag{A.12}$$

Using this rotation matrix, the new position of $\mathbf{X}$ is defined as follows:

$$\mathbf{X}' = R_y\mathbf{X} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X\cos(\theta) - Z\sin(\theta) \\ Y \\ X\sin(\theta) + Z\cos(\theta) \end{bmatrix}. \tag{A.13}$$

Since the left camera is at position $(b/2, 0, 0)$ of the 3D coordinate system, a translation is applied to the point as well, which yields,

$$\mathbf{X}'' = \begin{bmatrix} (X - b/2)\cos(\theta) - Z\sin(\theta) \\ Y \\ (X - b/2)\sin(\theta) + Z\cos(\theta) \end{bmatrix}. \tag{A.14}$$

Using the result of pinhole camera model, x_L and y_L are computed as follows:

$$x_L = f\frac{(X - b/2)\cos(\theta) - Z\sin(\theta)}{f - (X - b/2)\sin(\theta) - Z\cos(\theta)}, \tag{A.15}$$

$$y_L = f\frac{Y}{f - (X - b/2)\sin(\theta) - Z\cos(\theta)}. \tag{A.16}$$

Similarly x_R and y_R can be computed as:

$$x_R = f\frac{(X + b/2)\cos(\theta) + Z\sin(\theta)}{f + (X + b/2)\sin(\theta) - Z\cos(\theta)}, \tag{A.17}$$

$$y_R = f\frac{Y}{f + (X + b/2)\sin(\theta) - Z\cos(\theta)}, \tag{A.18}$$

and finally disparity is computed as:

$$\mathbf{d}_{LR} = \begin{bmatrix} x_L \\ y_L \end{bmatrix} - \begin{bmatrix} x_R \\ y_R \end{bmatrix} = \begin{bmatrix} fT_1/T_3 \\ fT_2/T_3 \end{bmatrix}, \tag{A.19}$$

where

$$T_1 = b(\cos^2(\theta)Z - \sin^2(\theta)Z - \cos(\theta)f) + (X^2 - b^2/4)\cos(\theta)\sin(\theta) + (2\cos(\theta)\sin(\theta)Z^2) - 2\sin(\theta)fZ, \tag{A.20}$$

$$T_2 = -2\sin(\theta)XY, \tag{A.21}$$

$$T_3 = (f - \cos(\theta)Z)^2 + b\sin(\theta)(f - \cos(\theta)Z) - \sin^2(\theta)(X^2 - b^2/4). \tag{A.22}$$

Note that when $\theta = 0$, (A.19) becomes equivalent to (A.8).

Appendix B

Euler-Lagrange equations for occlusion-aware optical flow estimation

Let $\Phi_L = \{u_L, v_L\}$, $\Phi_R = \{u_R, v_R\}$ be sets of disparity field components that we seek by minimizations in (6.12). Assuming that $\Omega_L = \Omega_R = \Omega$ in all energy terms (6.9–6.11), we can write (6.12) as follows:

$$E_L = \iint_{\Omega} e_L(\mathbf{x}) d\mathbf{x}, \tag{B.1}$$

$$E_R = \iint_{\Omega} e_R(\mathbf{x}) d\mathbf{x}, \tag{B.2}$$

where the integrands are defined as follows:

$$e_L(\mathbf{x}) = e_L^P(\mathbf{x}) + \eta e_L^S(\mathbf{x}) + \mu e_L^O(\mathbf{x}), \tag{B.3}$$

$$e_R(\mathbf{x}) = e_R^P(\mathbf{x}) + \eta e_S^R(\mathbf{x}) + \mu e_R^O(\mathbf{x}), \tag{B.4}$$

and where the individual terms are defined as follows:

$$\begin{aligned}
e_L^P(\mathbf{x}) &= D_L(\mathbf{x})[\rho_{LR}(\mathbf{x})]^2, \\
e_R^P(\mathbf{x}) &= D_R(\mathbf{x})[\rho_{RL}(\mathbf{x})]^2, \\
e_L^S(\mathbf{x}) &= F_{\mathbf{x}}(u_L, I_L) + F_{\mathbf{x}}(v_L, I_L), \\
e_R^S(\mathbf{x}) &= F_{\mathbf{x}}(u_R, I_R) + F_{\mathbf{x}}(v_R, I_R), \\
e_L^O(\mathbf{x}) &= (1 - D_L(\mathbf{x})), \\
e_R^O(\mathbf{x}) &= (1 - D_R(\mathbf{x})),
\end{aligned} \tag{B.5}$$

For simplicity let $D_L(\mathbf{x}) \stackrel{\triangle}{=} D(\epsilon_L(\mathbf{x}))$, $D_R(\mathbf{x}) \stackrel{\triangle}{=} D(\epsilon_R(\mathbf{x}))$.

We will minimize E_L and E_R simultaneously by assuming that $\mathbf{d}_L$ is constant when computing $\mathbf{d}_R$ and vice versa. This will lead to interleaved descent equations, i.e., one iteration of $\mathbf{d}_L$ using the values of $\mathbf{d}_R$ from previous iteration and vice versa.

Using the calculus of variations, two Euler-Lagrange equations (one for each unknown in $\Phi_\cdot$) for each $E_\cdot$ can be found in the form of

$$e'_L(\omega_L) = \frac{\partial e_L}{\partial \omega_L} - \frac{\partial}{\partial x}\frac{\partial e_L}{\partial \omega_L^x} - \frac{\partial}{\partial y}\frac{\partial e_L}{\partial \omega_L^y} = 0, \tag{B.6}$$

$$e'_R(\omega_R) = \frac{\partial e_R}{\partial \omega_R} - \frac{\partial}{\partial x}\frac{\partial e_R}{\partial \omega_R^x} - \frac{\partial}{\partial y}\frac{\partial e_R}{\partial \omega_R^y} = 0, \tag{B.7}$$

where $e'_L(w_L)$ and $e'_R(w_R)$ are the first variations with respect to $\omega_L \in \Phi_L$, $\omega_R \in \Phi_R$, whereas $\omega_\cdot^x$ and $\omega_\cdot^y$ are derivatives with respect to x and y, respectively. Expanding each equation and omitting derivatives that are equal to zero (e.g., $\partial e_L^P / \partial u_L^x = 0$) we get two Euler-Lagrange equations for each E as follows,

$$\begin{aligned} \frac{\partial e_L^P}{\partial \omega_L} + \mu\frac{\partial e_L^O}{\partial \omega_L} - \eta\left(\frac{\partial}{\partial x}\frac{\partial e_L^S}{\partial \omega_L^x} + \frac{\partial}{\partial y}\frac{\partial e_L^S}{\partial \omega_L^y}\right) &= 0, \\ \frac{\partial e_R^P}{\partial \omega_R} + \mu\frac{\partial e_R^O}{\partial \omega_R} - \eta\left(\frac{\partial}{\partial x}\frac{\partial e_R^S}{\partial \omega_R^x} + \frac{\partial}{\partial y}\frac{\partial e_R^S}{\partial \omega_R^y}\right) &= 0, \end{aligned} \tag{B.8}$$

where, again, $\omega_L \in \Phi_L$ and $\omega_R \in \Phi_R$. Partial derivatives with respect to u_L, v_L, u_R, v_R can be computed as follows ($\mathbf{x}$ was dropped for simplicity of notation, e.g., $\rho_{LR} \stackrel{\triangle}{=} \rho_{LR}(\mathbf{x})$):

$$\begin{aligned} \frac{\partial e_L^P}{\partial u_L} &= \frac{\partial D_L}{\partial u_L}(\rho_{LR})^2 - 2D_L\widetilde{I}_R^x\rho_{LR}, \\ \frac{\partial e_L^P}{\partial v_L} &= \frac{\partial D_L}{\partial v_L}(\rho_{LR})^2 - 2D_L\widetilde{I}_R^y\rho_{LR}, \\ \frac{\partial e_R^P}{\partial u_R} &= \frac{\partial D_R}{\partial u_R}(\rho_{RL})^2 - 2D_R\widetilde{I}_L^x\rho_{RL}, \\ \frac{\partial e_R^P}{\partial v_R} &= \frac{\partial D_R}{\partial v_R}(\rho_{RL})^2 - 2D_R\widetilde{I}_L^y\rho_{RL}, \end{aligned} \tag{B.9}$$

$$
\begin{aligned}
\frac{\partial}{\partial x}\frac{\partial e_L^S}{\partial u_L^x}+\frac{\partial}{\partial y}\frac{\partial e_L^S}{\partial u_L^y} &= \frac{\partial\left(2g(|I_L^x|)u_L^x\right)}{\partial x}+\frac{\partial\left(2g(|I_L^y|)u_L^y\right)}{\partial y},\\
\frac{\partial}{\partial x}\frac{\partial e_L^S}{\partial v_L^x}+\frac{\partial}{\partial y}\frac{\partial e_L^S}{\partial v_L^y} &= \frac{\partial\left(2g(|I_L^x|)v_L^x\right)}{\partial x}+\frac{\partial\left(2g(|I_L^y|)v_L^y\right)}{\partial y},\\
\frac{\partial}{\partial x}\frac{\partial e_R^S}{\partial u_R^x}+\frac{\partial}{\partial y}\frac{\partial e_R^S}{\partial u_R^y} &= \frac{\partial\left(2g(|I_R^x|)u_R^x\right)}{\partial x}+\frac{\partial\left(2g(|I_R^y|)u_R^y\right)}{\partial y},\\
\frac{\partial}{\partial x}\frac{\partial e_R^S}{\partial v_R^x}+\frac{\partial}{\partial y}\frac{\partial e_R^S}{\partial v_R^y} &= \frac{\partial\left(2g(|I_R^x|)v_R^x\right)}{\partial x}+\frac{\partial\left(2g(|I_R^y|)v_R^y\right)}{\partial y},
\end{aligned}
\tag{B.10}
$$

$$
\begin{aligned}
\frac{\partial e_L^O}{\partial u_L} &= -\frac{\partial D_L}{\partial u_L},\\
\frac{\partial e_L^O}{\partial v_L} &= -\frac{\partial D_L}{\partial v_L},\\
\frac{\partial e_R^O}{\partial u_R} &= -\frac{\partial D_R}{\partial u_R},\\
\frac{\partial e_R^O}{\partial v_R} &= -\frac{\partial D_R}{\partial v_R},
\end{aligned}
\tag{B.11}
$$

where $I_\cdot^x$ and $I_\cdot^y$ are horizontal and vertical derivatives of $I_\cdot$, while $\widetilde{I}_\cdot^x$ and $\widetilde{I}_\cdot^y$ are derivatives evaluated at a point off $\mathbf{x}$, e.g., $\widetilde{I}_L^x(\mathbf{x}) = I_L^x(\mathbf{x}+\mathbf{d}_R(\mathbf{x}))$. Furthermore, we have:

$$
\begin{aligned}
\frac{\partial D_L}{\partial u_L} &= -2K\frac{(1+\widetilde{u}_R^x)\epsilon_{L,u}+\widetilde{v}_R^x\epsilon_{L,v}}{(1+K(\epsilon_{L,u})^2+K(\epsilon_{L,v})^2)^2},\\
\frac{\partial D_L}{\partial v_L} &= -2K\frac{\widetilde{u}_R^y\epsilon_{L,u}+(1+\widetilde{v}_R^y)\epsilon_{L,v}}{(1+K(\epsilon_{L,u})^2+K(\epsilon_{L,v})^2)^2},\\
\frac{\partial D_R}{\partial u_R} &= -2K\frac{(1+\widetilde{u}_L^x)\epsilon_{R,u}+\widetilde{v}_L^x\epsilon_{R,v}}{(1+K(\epsilon_{R,u})^2+K(\epsilon_{R,v})^2)^2},\\
\frac{\partial D_R}{\partial v_R} &= -2K\frac{\widetilde{u}_L^y\epsilon_{R,u}+(1+\widetilde{v}_L^y)\epsilon_{R,v}}{(1+K(\epsilon_{R,u})^2+K(\epsilon_{R,v})^2)^2},
\end{aligned}
\tag{B.12}
$$

where

$$
\begin{aligned}
\epsilon_{L,u}(\mathbf{x}) &= u_L(\mathbf{x})+u_R(\mathbf{x}+\mathbf{d}_L(\mathbf{x})),\\
\epsilon_{L,v}(\mathbf{x}) &= v_L(\mathbf{x})+v_R(\mathbf{x}+\mathbf{d}_L(\mathbf{x})),\\
\epsilon_{R,u}(\mathbf{x}) &= u_R(\mathbf{x})+u_L(\mathbf{x}+\mathbf{d}_R(\mathbf{x})),\\
\epsilon_{R,v}(\mathbf{x}) &= v_R(\mathbf{x})+v_L(\mathbf{x}+\mathbf{d}_R(\mathbf{x})),
\end{aligned}
\tag{B.13}
$$

are individual components of disparity errors (6.8) and $\widetilde{u}_\cdot^x$, $\widetilde{v}_\cdot^x$, $\widetilde{u}_\cdot^y$, $\widetilde{v}_\cdot^y$ are again derivatives evaluated at a point off $\mathbf{x}$, e.g., $\widetilde{u}_L^x(\mathbf{x}) = u_L^x(\mathbf{x}+\mathbf{d}_R(\mathbf{x}))$. Using an auxiliary time variable

t, equations in (B.8) can be solved by discretizing the gradient descent equations

$$\frac{\partial \omega_L}{\partial t} = -e'_L(\omega), \tag{B.14}$$

$$\frac{\partial \omega_R}{\partial t} = -e'_R(\omega), \tag{B.15}$$

for $w_L \in \Phi_L$ and $w_R \in \Phi_R$.

Appendix C

Euler-Lagrange equations for occlusion-aware pivoting-based multiview method

We would like to minimize the following energy with respect to $\mathbf{d}$:

$$E = \int\int_{\mathbf{x}\in\Omega_J} e(\mathbf{x})d\mathbf{x}, \tag{C.1}$$

where $e(\mathbf{x}) = e_P(\mathbf{x}) + \lambda e_S(\mathbf{x})$. The integrands e_P and e_S are defined as follows:

$$e_P(\mathbf{x}) = P_{12}(\mathbf{x}) + P_{23}(\mathbf{x}) + P_{34}(\mathbf{x}), \tag{C.2}$$

$$e_S(\mathbf{x}) = F_{\mathbf{x}}(u_L, J_c) + F_{\mathbf{x}}(v_L, J_c), \tag{C.3}$$

where

$$P_{12}(\mathbf{x}) = \delta(L(\mathbf{x}) + 1)(\theta_{12}(\mathbf{x}))^2, \tag{C.4}$$

$$P_{23}(\mathbf{x}) = \delta(L(\mathbf{x}))(\theta_{23}(\mathbf{x}))^2, \tag{C.5}$$

$$P_{34}(\mathbf{x}) = \delta(L(\mathbf{x}) - 1)(\theta_{34}(\mathbf{x}))^2, \tag{C.6}$$

and

$$\theta_{12}(\mathbf{x}) = I_1(\mathbf{x} - (1+\alpha)\,\mathbf{d}(\mathbf{x})) - I_2(\mathbf{x} - \alpha\,\mathbf{d}(\mathbf{x})), \tag{C.7}$$

$$\theta_{23}(\mathbf{x}) = I_2(\mathbf{x} - \alpha\,\mathbf{d}(\mathbf{x})) - I_3(\mathbf{x} + (1-\alpha)\,\mathbf{d}(\mathbf{x})), \tag{C.8}$$

$$\theta_{34}(\mathbf{x}) = I_3(\mathbf{x} + (1-\alpha)\,\mathbf{d}(\mathbf{x})) - I_4(\mathbf{x} + (2-\alpha)\,\mathbf{d}(\mathbf{x})). \tag{C.9}$$

Using the calculus of variations, Euler-Lagrange equations for u and v can be found as follows:

$$\begin{aligned} e'(u) = &\quad \frac{\partial e}{\partial u} - \frac{\partial}{\partial x}\frac{\partial e}{\partial u^x} - \frac{\partial}{\partial y}\frac{\partial e}{\partial u^y} = 0, \\ e'(v) = &\quad \frac{\partial e}{\partial v} - \frac{\partial}{\partial x}\frac{\partial e}{\partial v^x} - \frac{\partial}{\partial y}\frac{\partial e}{\partial v^y} = 0, \end{aligned} \tag{C.10}$$

where u^x, v^x and u^y, v^y are horizontal and vertical derivatives of horizontal and vertical components of disparity. Expanding the equations, we get the Euler-Lagrange equations as follows:

$$\frac{e_P}{\partial u} - \frac{\partial}{\partial x}\frac{\partial e_S}{\partial u^x} - \frac{\partial}{\partial y}\frac{\partial e_S}{\partial u^y} = 0, \tag{C.11}$$

$$\frac{e_P}{\partial v} - \frac{\partial}{\partial x}\frac{\partial e_S}{\partial v^x} - \frac{\partial}{\partial y}\frac{\partial e_S}{\partial v^y} = 0. \tag{C.12}$$

Partial derivatives are defined as follows:

$$\frac{e_P}{\partial u} = \frac{\partial P_{12}}{\partial u} + \frac{\partial P_{23}}{\partial u} + \frac{\partial P_{34}}{\partial u}, \tag{C.13}$$

$$\frac{e_P}{\partial v} = \frac{\partial P_{12}}{\partial v} + \frac{\partial P_{23}}{\partial v} + \frac{\partial P_{34}}{\partial v}, \tag{C.14}$$

$$\frac{\partial}{\partial x}\frac{\partial e_S}{\partial u^x} + \frac{\partial}{\partial y}\frac{\partial e_S}{\partial u^y} = \frac{\partial\left(2g(|J_c^x|)u^x\right)}{\partial x} + \frac{\partial\left(2g(|J_c^y|)u^y\right)}{\partial y}, \tag{C.15}$$

$$\frac{\partial}{\partial x}\frac{\partial e_S}{\partial v^x} + \frac{\partial}{\partial y}\frac{\partial e_S}{\partial v^y} = \frac{\partial\left(2g(|J_c^x|)v^x\right)}{\partial x} + \frac{\partial\left(2g(|J_c^y|)v^y\right)}{\partial y}, \tag{C.16}$$

with

$$\frac{\partial P_{12}}{\partial u} = 2\delta(L(\mathbf{x})+1)\theta_{12}(\mathbf{x})\frac{\partial\theta_{12}(\mathbf{x})}{\partial u}, \tag{C.17}$$

$$\frac{\partial P_{12}}{\partial v} = 2\delta(L(\mathbf{x})+1)\theta_{12}(\mathbf{x})\frac{\partial\theta_{12}(\mathbf{x})}{\partial v}, \tag{C.18}$$

$$\frac{\partial P_{23}}{\partial u} = 2\delta(L(\mathbf{x}))\theta_{23}(\mathbf{x})\frac{\partial\theta_{23}(\mathbf{x})}{\partial u} \tag{C.19}$$

$$\frac{\partial P_{23}}{\partial v} = 2\delta(L(\mathbf{x}))\theta_{23}(\mathbf{x})\frac{\partial\theta_{23}(\mathbf{x})}{\partial v}, \tag{C.20}$$

$$\frac{\partial P_{34}}{\partial u} = 2\delta(L(\mathbf{x})-1)\theta_{34}(\mathbf{x})\frac{\partial\theta_{34}(\mathbf{x})}{\partial u}, \tag{C.21}$$

$$\frac{\partial P_{34}}{\partial v} = 2\delta(L(\mathbf{x})-1)\theta_{34}(\mathbf{x})\frac{\partial\theta_{34}(\mathbf{x})}{\partial v}, \tag{C.22}$$

where

$$\frac{\partial \theta_{12}(\mathbf{x})}{\partial u} = -(1+\alpha)\widetilde{I}_1^x + \alpha\widetilde{I}_2^x, \tag{C.23}$$

$$\frac{\partial \theta_{23}(\mathbf{x})}{\partial u} = -\alpha\widetilde{I}_2^x - (1-\alpha)\widetilde{I}_3^x, \tag{C.24}$$

$$\frac{\partial \theta_{34}(\mathbf{x})}{\partial u} = (1-\alpha)\widetilde{I}_3^x - (2-\alpha)\widetilde{I}_4^x, \tag{C.25}$$

$$\frac{\partial \theta_{12}(\mathbf{x})}{\partial v} = -(1+\alpha)\widetilde{I}_1^y + \alpha\widetilde{I}_2^y, \tag{C.26}$$

$$\frac{\partial \theta_{23}(\mathbf{x})}{\partial v} = -\alpha\widetilde{I}_2^y - (1-\alpha)\widetilde{I}_3^y, \tag{C.27}$$

$$\frac{\partial \theta_{34}(\mathbf{x})}{\partial v} = (1-\alpha)\widetilde{I}_3^y - (2-\alpha)\widetilde{I}_4^y, \tag{C.28}$$

where $I_\cdot^x$ and $I_\cdot^y$ are horizontal and vertical derivatives of $I_\cdot$, while $\widetilde{I}_\cdot^x$ and $\widetilde{I}_\cdot^y$ are derivatives evaluated at a point off $\mathbf{x}$, e.g., $\widetilde{I}_2^x = I_2^x(\mathbf{x} - \alpha\mathbf{d}(\mathbf{x}))$.

Using an auxiliary time variable t, equations in (C.10) can be solved by discretizing the gradient descent equations

$$\frac{\partial u}{\partial t} = -e'(u), \tag{C.29}$$

$$\frac{\partial v}{\partial t} = -e'(v). \tag{C.30}$$

References

Adelson, E. H. and Bergen, J. R. (1991). The plenoptic function and the elements of early vision. *M. Landy and J. A. Movshon, (eds) Computational Models of Visual Processing*.

Alatan, A. and Onural, L. (1998). Estimation of depth fields suitable for video compression based on 3-D structure and motion of objects. *IEEE Transactions on Image Processing*, 7(6):904–908.

Alvarez, L., Deriche, R., Papadopoulo, T., and Sánchez, J. (2002a). Symmetrical dense optical flow estimation with occlusions detection. In *Proceedings European Conference Computer Vision*, volume 1, pages 721–736.

Alvarez, L., Deriche, R., Sánchez, J., and Weickert, J. (2002b). Dense disparity map estimation respecting image discontinuities : A PDE and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13:3–21.

Alvarez, L., Esclarín, J., Lefébure, M., and Sánchez, J. (1999). A PDE model for computing the optical flow. In *Proceedings XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356.

Avidan, S. and Shashua, A. (1997). Novel view synthesis in tensor space. In *Proceedings of IEEE Conference Computer Vision Pattern Recognition*, pages 1034–1040.

Bell III, J. F. and *et al.* (2003a). Mars exploration rover athena panoramic camera (Pancam) investigation. *Journal of Geophysical Research*, 108(E12).

Bell III, J. F. and *et al.* (2003b). Pancam: A multispectral imaging investigation on the NASA 2003 Mars exploration rover mission. *Sixth International Conference on Mars*.

Benton, S. A., editor (2001). *Selected papers on three-dimensional displays*. SPIE Optical Engineering Press, Bellingham, WA.

Benyon, M. (1998). Prehistory of holographic art: a personal view. In *SPIE Sixth International Symposium on Display Holography*.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *SIGGRAPH'00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*.

Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision Image Understanding*, 63(1):75–104.

Buehler, C., Bosse, M., McMillan, L., Gortler, S., and Cohen, M. (2001). Unstructured lumigraph rendering. In *SIGGRAPH'01: Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 425–432.

Chahine, M. and Konrad, J. (1995). Estimation and compensation of accelerated motion for temporal sequence interpolation. *Signal Processing: Image Communication*, 7(4–6):503–527.

Chen, S. E. and Williams, L. (1993). View interpolation for image synthesis. In *SIGGRAPH'93: Proceedings of the 20th annual conference on computer graphics and interactive techniques*, pages 279–288.

Debevec, P. E., Borshukov, G., and Yu, Y. (1998). Efficient view-dependent image-based rendering with projective texture-mapping. In *9th Eurographics Rendering Workshop*.

Dempster, A. P., Laird, N. M., and Rubin, D. P. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statics Society*, B 39(1):1–38.

Depommier, R. and Dubois, E. (1992). Motion estimation with detection of occlusion areas. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 269–272.

Dodgson, N. (2005). Autostereoscopic 3D displays. *Computer*, 38(8):31–36.

Driessen, J. and Biemond, J. (1991). Motion field estimation for complex scenes. In *Proceedings SPIE Visual Communications and Image Processing*, pages 511–521.

Ellis, S. R., Fishman, J. M., Hasser, C. J., and Stern, J. D. (2005). Effect of reduced stereoscopic camera separation on ring placement with a surgical telerobot. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 372–379.

Faugeras, O. (1993). *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA.

Franich, R. (1996). *Disparity estimation in stereoscopic digital images*. PhD thesis, Delft University of Technology.

Geiger, B., Ladendorf, B., and Yuille, A. (1995). Occlusions and binocular stereo. *International Journal of Computer Vision*, 14:211–226.

Gennert, M. and Yuille, A. (1988). Determining the optimal weights in multiple objective function optimization. In *Proceedings of IEEE International Conference Computer Vision*, volume 2, pages 87–89.

Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The lumigraph. In *SIGGRAPH'96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, pages 43–54.

Grimson, W. E. L. (1985). Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 7:17–34.

Halle, M. (1997). Autostereoscopic displays and computer graphics. *Computer Graphics ACM SIGGRAPH*, 31(2):58–62.

Hartley, R. and Zisserman, A. (2004). *Multi view geometry in computer vision*. Cambridge University Press.

Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Howard, I. P. and Howard, A. P. (1995). *Binocular Vision and Stereopsis*. Oxford University Press.

Huang, X. and Dubois, E. (2005). Three-view dense disparity estimation with occlusion detection. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 393–396.

Hubbold, R. J., Hancock, D. J., and Moore, C. J. (1997). Autostereoscopic display for radiotherapy planning. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 16–27.

IJsselsteijn, W. A., de Ridder, H., and Vliegen, J. (2000). Subjective evaluation of stereoscopic images: Effects of camera parameters and display duration. *IEEE Transactions on Circuits Systems and Video Technology*, 10(2):225–233.

Ilgner, J., Kawai, T., Westhofen, M., and Shibata, T. (2004). Production and evaluation of stereoscopic video presentation in surgical training. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 293–302.

Ince, S. and Konrad, J. (2005a). Geometry-based estimation of occlusions from video frame pairs. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 933–936.

Ince, S. and Konrad, J. (2005b). Recovery of a missing color component in stereo images (or helping NASA find little green martians). In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, volume 5664, pages 127–138.

Ince, S. and Konrad, J. ((submitted) 2007). Occlusion-aware optical flow estimation. *IEEE Transactions on Image Processing*.

Ince, S., Konrad, J., and Vázquez, C. (2007a). Spline-based intermediate view reconstruction. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, volume 6490, pages 0F.1–0F.12.

Ince, S., Martinian, E., Yea, S., and Vetro, A. (2007b). Depth estimation for view synthesis in multiview video coding. In *Proceedings of IEEE 3D TV Conference*.

Iu, S.-I. (1995). Robust estimation of motion vector fields with discontinuity and occlusion using local outliers rejection. *Journal of Visual Communications and Image Representation*, 6(2):132–141.

Izquierdo, E. (1997). Stereo matching for enhanced telepresence in three- dimensional videocommunications. *IEEE Transactions on Circuits Systems and Video Technology*, 7(4):629–643.

JM (2006). H.264/AVC JM Reference Software. `iphome.hhi.de/suehring/tml`.

JTC1/SC29/WG11, I. (2005). Updated call for proposals on multi-view video coding. N7567. Nice, France.

Kanade, T., Rander, P., and Narayanan, P. J. (1997). Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47.

Kardouchi, M. and Konrad, J. (2003). Recovering large-amplitude disparity fields using adaptive interpolation. In *Proceedings SPIE Image and Video Communications and Processing*, volume 5022, pages 761–771.

Karl, W. C. (2005). Regularization in image restoration and reconstruction. In Bovik, A., editor, *Handbook of Image and Video Processing*. Academic Press.

Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160.

Kim, H. and Sohn, K. (2005). 3D reconstruction from stereo images for interactions between real and virtual objects. *Signal Processing: Image Communication*, 20(1):61–75.

Kimata, H. and Kitahara, M. (2004). Preliminary results on multiple view video coding. ISO/IEC JTC1/SC29/WG11, M10976.

Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. In *Proceedings of IEEE International Conference Computer Vision*, volume 2, pages 508–515.

Konrad, J. (1999). Enhancement of viewer comfort in stereoscopic viewing: parallax adjustment. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, volume 3639, pages 179–190.

Konrad, J. (2001). Visual communications of tomorrow: natural, efficient and flexible. *IEEE Communication Magazine*, 39(1):126–133.

Konrad, J. and Agniel, P. (2006). Subsampling models and anti-alias filters for 3-D automultiscopic displays. *IEEE Transactions on Image Processing*, 15(1):128–140.

Konrad, J. and Božinović, N. (2005). Importance of motion in motion-compensated temporal discrete wavelet transforms. In *Proceedings SPIE Image and Video Communications and Processing*, volume 5685, pages 354–365.

Konrad, J. and Halle, M. (2007). 3-D displays and signal processing: An answer to 3-D ills? *IEEE Signal Processing Magazine*, pages 97–111.

Levoy, M. and Hanrahan, P. (1996). Light field rendering. In *SIGGRAPH'96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, pages 31–42.

Lim, K., Das, A., and Chong, M. (2002). Estimation of occlusion and dense motion fields in a bidirectional Bayesian framework. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24(5):712–718.

Lipton, L. (1991). *StereoGraphics Developer's Handbook*. StereoGraphics Corporation.

Lipton, L. (1994). True stereoscopic television: 3D-TV is feasible and striking. *Advanced Imaging*, pages 28–30.

Mancini, A. and Konrad, J. (1998). Robust quadtree-based disparity estimation for the reconstruction of intermediate stereoscopic images. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, volume 3295, pages 53–64.

Mansouri, A.-R. and Konrad, J. (2000). Bayesian winner-take-all reconstruction of intermediate views from stereoscopic images. *IEEE Transactions on Image Processing*, 9(10):1710–1722.

Mansouri, A.-R., Mitiche, A., and Konrad, J. (1998). Selective image diffusion: application to disparity estimation. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 284–288.

March, R. (1988). Computation of stereo disparity using regularization. *Pattern Recognition Letters*, 8:181–187.

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, 194:283–287.

Martinian, E., Behrens, A., Xin, J., and Vetro, A. (2006a). View synthesis for multiview video compression. In *Picture Coding Symposium*.

Martinian, E., Behrens, A., Xin, J., Vetro, A., and Sun, H. (2006b). Extensions of H.264/AVC for multiview video compression. In *Proceedings of IEEE International Conference on Image Processing*.

Matusik, W., Buehler, C., Raskar, R., Gortler, S., and McMillan, L. (2001). Image-based visual hulls. In *SIGGRAPH'00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*, pages 369–374.

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

McAllister, D. F. (1993). *Stereo Computer Graphics and Other True 3D Technologies*. Princeton University Press.

McMillan, L. (1997). *An image-based approach to three-dimensional computer graphics*. PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

McVeigh, J., Siegel, M., and Jordan, A. (1996). Intermediate view synthesis considering occluded and ambiguously referenced image regions. *Signal Processing: Image Communication*, 9:21–28.

Merkle, P., Muller, K., Smolic, and Wiegand, T. (2006). Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG4-AVC. In *IEEE Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1717–1720.

Moravec, H. (1980). *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford University.

Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 8(5):565–593.

Ng, L. and Solo, V. (1997). A data-driven method for choosing smoothing parameters in optical flow problems. In *Proceedings of IEEE International Conference on Image Processing*, pages 360–363.

Oddsson, L., Karlsson, R., Konrad, J., Ince, S., Williams, S., and Zemkova, E. (2007). A rehabilitation tool for functional balance using altered gravity and virtual reality. *Journal of NeuroEngineering and Rehabilitation*, 4 (25).

Oddsson, L., Konrad, J., Williams, S., Karlsson, R., and Ince, S. (2006). A rehabilitation tool for functional balance using altered gravity and virtual reality. In 5^{th} *International Workshop on Virtual Rehabilitation*.

Oddsson, L., Wall III, C., Meyer, P., and Konrad, J. (2004). A virtual environment with simulated gravity for balance rehabilitation of bedridden patients and frail individuals. In *XV-th Congress of the International Society of Electrophysiology and Kinesiology*, page 55.

Olson, C. F., Matthies, L. H., Schoppers, M., and Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229.

Papadimitriou, D. and Dennis, T. (1996). Epipolar line estimation and rectification for stereo image pairs. *IEEE Transactions on Image Processing*, 5(4):672–676.

Park, J. and Inoue, S. (1997). Arbitrary view generation using multiple cameras. In *Proceedings of IEEE International Conference on Image Processing*, volume 1, pages 149–153.

Park, J. and Park, H. (2006). A mesh-based disparity representation method for view interpolation and stereo image compression. *IEEE Transactions on Image Processing*, 15(7):1751–1762.

Pastoor, S. and Wöpking, M. (1997). 3-D displays: A review of current technologies. *Displays*, 17:100–110.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 12(7): 629–639.

Pollard, S. B., Mayhew, J. E., and Frisby, J. P. (1985). PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14(4):449–470.

Proesmans, M., Gool, L. V., Pauwels, E., and Oosterlinck, A. (1994). Determination of optical flow and its discontinuities using non-linear diffusion. In *3rd Eurpoean Conference on Computer Vision*, volume 2, pages 295–304.

Redert, A., de Beeck, M., Fehn, C., Ijsselsteijn, W., Pollefeys, M., Gool, L., Ofek, E., Sexton, I., and Surman, P. (2002). ATTEST: Advanced three-dimensional television system technologies. In *Proceedings of International Symposium on 3D Data Processing Visualization And Transmission*, pages 313–319.

Redert, A., Hendriks, E., and Biemond, J. (1997). Synthesis of multi viewpoint images at non-intermediate positions. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume IV, pages 2749–2752.

Ristivojević, M. and Konrad, J. (2004). Joint space-time image sequence segmentation: object tunnels and occlusion volumes. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume III, pages 9–12.

Robert, L. and Deriche, R. (1996). Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In *Proceedings European Conference Computer Vision*, volume I, pages 439–451.

Salb, T., Brief, J., Burgert, O., Hassfeld, S., and Dillmann, R. (2000). Intraoperative presentation of surgical planning and simulation results using a stereoscopic see-through head-mounted display. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 68–75.

Salimpour, P., Kim, C. A., LaMorte, W., Birkett, D., and Babayan, R. K. (1997). Comparison of a new glasses-free three dimensional screen, a passive- glasses three dimensional screen and a two-dimensional imaging system for use in laparoscopic surgery. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 7–15.

Scharstein, D. (1996). Stereo vision for view synthesis. In *Proceedings of IEEE Conference Computer Vision Pattern Recognition*.

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

Seitz, S. and Dyer, C. (1996). View morphing. In *SIGGRAPH'96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, pages 21 – 30.

Shum, H., Kang, S., and Chan, S. (2003). Survey of image-based representations and compression techniques. *IEEE Transactions on Circuits Systems and Video Technology*, 13(11):1020–1037.

Siu, A. and Lau, R. (2005). Image registration for image-based rendering. *IEEE Transactions on Image Processing*, 14(2):1057–7149.

Smolic, A. and Kimata, H. (2003). Report on 3DAV exploration. ISO/IEC JTC1/SC29/WG11, N5878.

Strecha, C. and Gool, L. V. (2002). PDE-based multi-view depth estimation. In *1st International Symposium of 3D Data Processing Visualization and Transmission*, volume 2, pages 416–425.

Sullivan, A. (2005). 3-Deep: New displays render images you can almost reach out and touch. *IEEE Spectrum*, 42(4):30–35.

Sun, J., Li, Y., Kang, S. B., and Shum, H.-Y. (2005). Symmetric stereo matching for occlusion handling. In *Proceedings of IEEE Conference Computer Vision Pattern Recognition*, volume 2.

Thoma, R. and Bierling, M. (1989). Motion compensating interpolation considering covered and uncovered background. *Signal Processing: Image Communication*, 1:191–212.

Unser, M. (1999). Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38.

Vázquez, C., Dubois, E., and Konrad, J. (2005). Reconstruction of irregularly-sampled images in spline spaces. *IEEE Transactions on Image Processing*, 14(6):713–725.

Vese, L. and Osher, S. (2003). Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19(1-3):553–572.

Vetro, A., Matusik, W., Pfister, H., and Xin., J. (2004). Coding approaches for end-to-end 3D TV systems. In *Picture Coding Symposium*.

Wang, X. H., Good, W. F., Fuhrman, C. R., Sumkin, J. H., Britton, C. A., Warfel, T. E., and Gur, D. (2004). Stereo display for chest CT. In *Proceedings SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 17–24.

Weickert, J. and Schnörr, C. (2001). A theoretical framework for convex regularizers in PDE-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264.

Wheatstone, C. (1838). On some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394.

Wilburn, B., Joshi, N., Vaish, V., Talvala, E.-V., Antunez, E., Barth, A., Adams, A., Horowitz, M., and Levoy, M. (2005). High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24(3):765–776.

Wilburn, B., Smulski, M., Lee, H.-H. K., and Horowitz, M. (2002). Light field video camera. In *Proceedings SPIE Media Processors*, volume 4674, pages 29–36.

Woods, A., Docherty, T., and Koch, R. (1993). Image distortions in stereoscopic video systems. In *Proceedings SPIE Stereoscopic Displays and Applications*, volume 1915.

Xiao, J., Cheng, H., Sawhney, H., Rao, C., and Isnardi, M. (2006). Bilateral filtering-based optical flow estimation with occlusion detection. In *Proceedings European Conference Computer Vision*.

Xiao, J. and Shah, M. (2005). Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 27(10):1644–1659.

Zhai, J., Yu, K., Li, J., and Li, S. (2005). A low complexity motion compensated frame interpolation method. In *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 5, pages 4927– 4930.

Zhang, C. and Chen, T. (2004). A survey on image-based rendering–representation, sampling and compression. *Signal Processing: Image Communication*, 19(1):1–28.

Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R. (2004). High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608.

CURRICULUM VITAE

Serdar İnce

Contact Information

Address: ECE Department 8 St. Mary's Street Boston MA 02215
Email: ince@bu.edu
Web: http://iss.bu.edu/ince

Education

Ph.D., Boston University, 2008 (expected)
Department of Electrical and Computer Engineering

M.S., Middle East Technical University, 2002
Department of Electrical and Electronics Engineering

B.S., Middle East Technical University, 2000
Department of Electrical and Electronics Engineering

Work Experience

Research Assistant May 2003 - current
Visual Information Laboratory, Boston University, Boston MA

Intern Mar 2006 - Sep 2006
Mitsubishi Electric Research Laboratories, Cambridge MA

Research Assistant May 2005 - Sep 2007
NeuroMuscular Research Center, Boston University, Boston MA

Teaching Assistant Sep 2002 - May 2002
Department of Electrical and Computer Eng., Boston University, Boston MA

Research Assistant Sep 2000 - Aug 2002
Multimedia Research Group, Middle East Technical University, Ankara Turkey

Honors

- Runner up prize for presentation in SDA'05 (Jan 2005)
- Bülent Kerim Altay Outstanding Success Award (Jun 2000)

Publications

Refereed Publications:

- S. Ince and J. Konrad "Occlusion-aware optical flow estimation," in IEEE Trans. on Image Processing (submitted)

- L. Oddsson, R. Karlsson, J. Konrad, S. Ince, S. Williams, and E. Zemkova, "A rehabilitation tool for functional balance using altered gravity and virtual reality," Journal of NeuroEngineering and Rehabilitation, Jul. 2007
- S. Ince, E. Martinian, S. Yea and A. Vetro, "Depth estimation for view synthesis in multiview video coding," in Proc. of IEEE 3D TV Conference, May 2007
- S. Ince, J. Konrad, and C. Vazquez, "Spline-based intermediate view reconstruction," in Proc. SPIE Stereoscopic Displays and Virtual Reality Systems, Jan. 2007
- R. Lau, S. Ince, and J. Konrad, "Compression of still multi-view images for 3-D automultiscopic spatially-multiplexed displays," in Proc. SPIE Stereoscopic Displays and Virtual Reality Systems, Jan. 2007
- L. Oddsson, J. Konrad, S.R. Williams, R. Karlsson and S. Ince, "A rehabilitation tool for functional balance using altered gravity and virtual reality," in Proc. of 5th International Workshop of Virtual Rehabilitation, Aug. 2006
- S. Ince and J. Konrad, "Geometry-based estimation of occlusions from video frame pairs," in Proc. IEEE International Conference Acoustics Speech and Signal Processing, Mar. 2005
- S. Ince and J. Konrad, "Recovery of a missing color component in stereo images (or helping NASA find little green martians)," in Proc. SPIE Stereoscopic Displays and Virtual Reality Systems, Jan. 2005
- S. Ince, E. Gurses and G. Bozdagi Akar, "A real-time multimedia communication application for serial channels (Seri kanallar icin gercek zamanli cogulortam haberlesmesi uygulamasi)," in Proc. IEEE National Conference on Signal Processing and Applications, 2002 (in Turkish)
- S. Ince and G. Bozdagi Akar, "Implementation of a video encoder on a digital signal processor for low-bitrate communication (Sayisal sinyal islemcisi uzerinde dusuk hizli iletisim icin video kodlamasinin gerceklemesi)," in Proc. ITUS 2001 (in Turkish)

Patent:

- 1 patent pending in the area of multiview video coding

MPEG Contributions:

- S. Ince, E. Martinian, S. Yea, A. Vetro, "Preliminary results on CE 3: view synthesis for multiview video", ISO/IEC JTC1/SC29/WG11, M13123, Montreux, Switzerland, April 2006
- S. Yea, J. Oh, S. Ince, E. Martinian, A. Vetro, "Results on CE 3B: view synthesis for multiview video", ISO/IEC JTC1/SC29/WG11, Klagenfurt, Austria, July 2006

Thesis:

- S. Ince, "Implementation of a real-time video codec on a multi processor DSP architecture", M.S. Thesis, METU, Jun 2002