INTERMEDIATE VIEW RECONSTRUCTION FOR MULTISCOPIC 3D DISPLAY

KARAJEH, HUDA,ABDEL-RAHIM

INTERMEDIATE VIEW RECONSTRUCTION FOR MULTISCOPIC 3D DISPLAY

Huda A. Karajeh

A Thesis presented for the degree of

Doctor of Philosophy

School of Engineering and Computing Sciences

Durham University

United Kingdom

2011

To my Father (Abdel-Raheem Karajeh) who passed away during my PhD Study.

May Allah rest his gentle soul in eternal peace,

Aamen

Abstract

This thesis focuses on Intermediate View Reconstruction (IVR) which generates additional images from the available stereo images. The main application of IVR is to generate the content of multiscopic 3D displays, and it can be applied to generate different viewpoints to Free-viewpoint TV (FTV). Although IVR is considered a good approach to generate additional images, there are some problems with the reconstruction process, such as detecting and handling the occlusion areas, preserving the discontinuity at edges, and reducing image artifices through formation of the texture of the intermediate image. The occlusion area is defined as the visibility of such an area in one image and its disappearance in the other one. Solving IVR problems is considered a significant challenge for researchers.

In this thesis, several novel algorithms have been specifically designed to solve IVR challenges by employing them in a highly robust intermediate view reconstruction algorithm. Computer simulation and experimental results confirm the importance of occluded areas in IVR. Therefore, we propose a novel occlusion detection algorithm and another novel algorithm to Inpaint those areas. Then, these proposed algorithms are employed in a novel occlusion-aware intermediate view reconstruction that finds an intermediate image with a given disparity between two input images. This novelty is addressed by adding occlusion awareness to the reconstruction algorithm and proposing three quality improvement techniques to reduce image artifices: filling the re-sampling holes, removing ghost contours, and handling the disocclusion area.

We compared the proposed algorithms to the previously well-known algorithms on each field qualitatively and quantitatively. The obtained results show that our algorithms are superior to the previous well-known algorithms. The performance of the proposed reconstruction algorithm is tested under 13 real images and 13 synthetic images. Moreover, analysis of a human-trial experiment conducted with 21 participants confirmed that the reconstructed images from our proposed algorithm have very high quality compared with the reconstructed images from the other existing algorithms.

Declaration

The work in this thesis is based on research carried out at the Innovative Computing Group, School of Engineering and Computing Sciences, Durham University. No part of this thesis has been submitted elsewhere for any other degree or qualification and all of it is my own work unless referenced to the contrary in the text.

Acknowledgement

Firstly, all thanks to Allah who gave me the ability to finish this work successfully; without the mercy of Allah, I could not achieve anything. Then I would like to thank my Father, who passed away during my PhD study, for his support and encouragement to continue my study. I know his soul is very happy about my successful achievement. Then I would like to thank my Mother, who is always supporting me and praying for me to obtain this degree successfully. I respect her patience during my absence from her all these years. Special thanks go to my Mother-in-law and Father-in-law for their love and support throughout my studies.

Then, I would like to thank my supervisor, Dr. Nick Holliman, for giving me the guidance and motivation that was needed to lead me in the correct direction, Prof. Frank Coolen, for his very helpful advice on statistics, and Paul Gorley for his help in the human-trial experiment. Many thanks and grateful acknowledgement for proofreading and correcting the English edition go to John R. Coast.

I send my gratitude and love to my husband Mahmoud, who encouraged me to study for this PhD, which has greatly influenced my life. Without his endless patience, understanding, support and help, I would never have been able to finish this work. I would like to give special thanks to my cute son Ayham who came into this life while I was studying for my PhD. His gorgeous smile blessed and encouraged me to continue my studies. My studies may have kept me from him a little bit, but he will be proud when he realizes what I have achieved. Special thanks go to my sister Nuha and her husband Amjad, who supported me a lot to study for this PhD; without their support it would not have been possible for me to achieve this degree. I send many thanks to my sister Ola who spent a lot of time encouraging me when I went through tough times. I express my gratitude to all my sisters and brothers for their support.

I am very grateful to the Department of Computer Science at Durham University for the availability of the Visualization Lab with its invaluable and expensive displays, which enabled us to evaluate this work qualitatively. Finally, I would like to thank my sponsor, the University of Jordan, which funded me to complete my studies.

Huda Karajeh

Table of Content

List of Figures

List of Tables

CHAPTER

1

Introduction

Recently, the multiview 3D display has been developed to overcome the deficiencies of stereoscopic systems. These displays need to generate multiple images from the available stereo image pairs in the stereoscopic system. This generation process is called Intermediate View Reconstruction (IVR). Although it is considered a good approach for generating multiple images, there are some problems through the reconstruction process such as finding a reliable disparity map, handling the occlusion areas and preserving the discontinuity of edges. Solving these problems is considered a significant challenge for researchers. Thus, the main aim of this thesis is to reconstruct an intermediate image while detecting and handling the occlusion areas, preserving the discontinuity at edges, and reducing image artifices through formation of the texture of the intermediate image.

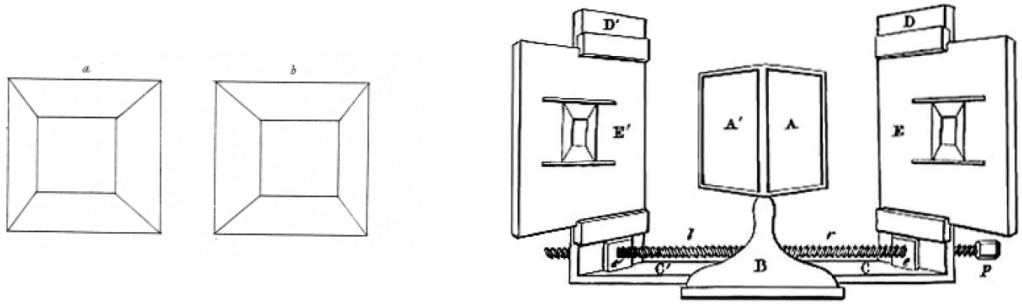
1.1 Introduction

Thanks to the recent advances in media technology, many devices and displays have been developed to simulate the stereo vision of the human eyes, allowing the viewer to perceive depth. In computer vision, the human eyes are replaced by two cameras separated by the same distance between the human eyes and then acquiring two different images; these two images are called “stereo images” and this system is called the *stereoscopic* system. When these two images are delivered to the viewer’s eyes, each eye receives a corresponding image, and the viewer will perceive depth. This depth is generated when each eye sees a separate 2D image of the same scene from different perspective; then the brain extracts the difference between these images to generate the perceived depth. As a next step, it is

worth knowing how the idea of a stereoscopic system was formed and by whom; this is introduced in the next section.

1.2 A Brief History of the Stereoscopic System

The concept of 3D was established a long time ago, amazingly, before the invention of photography. The scientist and artist Leonardo da Vinci (1452-1519) studied the concept of depth perception, according to his notes and drawings, when he tried to understand the difference between how humans see a scene in the real world and how it is painted [1]. He subsequently concluded that it is impossible to perceive depth as a human perception from a single picture [2]. This concept remained ambiguous for 300 years until the physicist Charles Wheatstone (1802-1875) invented the first stereoscopic device. He explained this invention in his publication in June,1838 [3] and presented it to the Royal Society. Wheatstone explained in his article that the perception of depth is achieved by showing each eye a different 2D image of the same scene but from different viewpoints. These two images must be different only in the horizontal coordinate to imitate the human binocular vision. As photography had yet to be invented, Wheatstone used simple hand drawings to demonstrate his newly discovered device, which is called “Mirror Stereoscope” and is now preserved in the Science Museum in London. Figure 1.1 (a) shows one of the hand drawings used by Wheatstone’s stereoscopic device which is also shown in Figure 1.1 (b). As shown in the Figure 1.1, the Wheatstone stereoscope used angled mirrors A and A’ to reflect the stereoscopic drawings E and E’ toward the viewer’s eyes.



(a) (b)

Figure 1.1: (a) Wheatstone’s hand drawing (b) Wheatstone’s stereoscopic device

After Wheatstone's invention, Dr. David Brewster (1781-1868) greatly influenced the stereoscopic field with his new stereoscope in 1849; this is called the Lenticular Stereoscope. This new stereoscope is completely different from the Wheatstone devices. After that, a company started to manufacture this stereoscope and exhibited at the Great Exhibition in London in 1851. Queen Victoria visited this exhibition and listened in wonder as Dr. Brewster explained this stereoscope to her. Within a few months, 250,000 stereoscopes had been sold, thus representing the first widespread use of stereoscopic devices [4]. Since the first stereoscopic device invention, many new technologies have emerged. In the following chapter, we will cover the main developments in stereoscopic devices.

1.3 Stereoscopic Benefits and Applications

Due to the importance of the presence of depth in a scene, its employment on a computer can be exploited in many applications and in different areas such as in entertainment, medicine, scientific research, virtual reality, business etc. Some of the benefits of binocular vision [5] are summarized as follows:

- *Sense of depth*: this depth can be used to determine the relative distance of objects from the viewer.
- *Analyzing complex data more efficiently*: since binocular vision allows humans to focus on certain objects and ignore others [5], presenting a pair of stereo images on a 3D display will improve the understanding of a complex scene due to the viewer's ability to separate the object from the background [6].
- *Binocular unmasking*: binocular vision allows humans to easily distinguish the desired object in a noisy environment [7]. Thus, a noisy pair of stereo images in a 3D display will be seen more clearly than a single 2D image with the same degree of noise. So, the object can be easily detectable in a noisy scene by separating the noise from the object when presenting a pair of stereo images. Thus, presenting depth in a 3D scene is not hampered by the presence of image noise in the stereo images as the retinal disparity cues are not highly dependent on the image quality [8].

By demonstrating stereoscopic benefits, stereoscopic images can be exploited in many applications. Entertainment is one the major applications for stereoscopic displays, since there is no doubt that the presence of depth in images gives the viewer a more pleasant experience. Therefore, 3D cinema has become popular – for example, the IMAX TM movie theatres - since the first 3D cinema movie was produced in 1966 [9]. For a long time, researchers have been working hard to generate a satisfactory 3D TV; this may be the next generation of TVs and may replace conventional ones [10-12]. Recently, 3D technology has been playing its part in video games. Adding depth to the video games allows the player to become more excited and enthusiastic about a game. 3D video games can be run on 3D displays of a PC or Laptop, 3DTV, 3D mobile, or on specialized games devices; for example, the Nintendo Inc. developed a 3D DS console that does not require the player to wear glasses.

Since a stereoscopic imaging system enables complex data to be more understandable and clear for the viewer, it is suitable for use in scientific visualisations to analyze data. It is also considered useful in remote guidance, especially in tasks that are hazardous for humans, as well as in medical surgery. Furthermore, as a stereoscopic system is useful in breaking camouflage that is extended from binocular unmasking, it can be employed in underwater tasks that suffer from a turbid environment and low illumination [13].

Videoconferencing is one of the more recent uses of autostereoscopic system application, since it creates an interactive communication between the remote conferees. Researchers are trying to make this contact seem as real as possible by entering the world of virtual reality; this can be done by trying to imitate the same capability of the real contact. In reality, if one of the conferees move their head, different viewpoints will be seen. So, two stereo images for the same scene are not enough to make the remote appear natural, because the conferee will see the other end from one viewpoint; this conflicts with reality. Therefore, videoconferencing is done via a multiview autostereoscopic system. Thus, a good multiview display and software for generating the intermediate images between the stereo pair (i.e. the multiview images), with high quality is required.

1.4 The Reality of Stereoscopy

Since the main goal of a stereoscopic system is to imitate the human ability to see the world in 3D, an essential question must be answered: Does the stereoscopic system achieve a real 3D, and to what degree does it look like the natural viewing capability? By introducing the main characteristics of a real viewing capability, this question will be answered.

1.4.1 The Realistic Aspects of Viewing 3D

The presence of depth alone in a scene is not enough to give a natural viewing capability, so two main aspects must also be available while watching a 3D scene.

1. *Natural look-around capability*:

In the real world, if the viewer's head moves, and different images will be seen; this is called motion parallax which in turn let us see around the object. Thus, motion parallax gives us the look-around capability that can be defined as the ability to see different perspectives of the same scene from different positions. Thus, there is no system that gives a real viewing experience without the availability of motion parallax. Since stereoscopic systems try to generate depth by seeing a separate 2D image in each eye, the viewer sees just one fixed scene from different positions; therefore a stereoscopic system cannot achieve real 3D.

The solution is to generate multiple images of the same scene from different perspectives. Technologies to display these multiple images are available, but the process of generating these images is considered a hard task and is not without problems.

2. Comfortable viewing:

Since a stereoscopic system is devised to simulate the stereo vision of the human eyes, they will receive via a stereoscopic system two images supplied by two cameras separated by a distance equal to the distance between the human eyes. However, do all humans have the same distance between their eyes? If not, what would be a suitable separation distance to select? Unfortunately, human eye

separation varies between 40 mm and 80 mm [14]. By this, the perception of depth present in the view is dependent on the viewing ability of the observer [15], in that each observer can see a different degree of depth. The availability of just two images in the stereoscopic images doesn't permit the viewer to perceive the preferred depth, thus obligating him/her to converge and accommodate his/her eyes according to the depth presented in the scene. Consequently, this will cause eye strain and visual discomfort to the viewer. This discomfort is due to the inconsistency between accommodation and convergence of the viewer's eyes, especially when the depth presented in the scene is larger than the preferred depth. Therefore, the researchers have agreed to separate the cameras by the average human eye separation (65 mm) [14], which is not a comfortable solution for all viewers, especially if the observer has an eye separation far from the average; for example 80 .

Presenting multiple images of the same scene but from different perspectives will allow the viewer to select a pair of images with the preferred sense of depth for comfortable viewing.

1.4.2 The Reality and Multiview Systems: Thesis Motivations

As was mentioned above, the reality of a stereoscopic system is achieved by presenting the two essential elements of realism: the *"natural look-around"* capability, also called *"continuous motion parallax"*, and comfortable viewing. One suggestion for achieving this is to present multiple images of the same scene from different perspectives to the viewer, thus presenting different pairs of views as the viewer's head moves. In this way, the viewer can select a suitable pair of views that deliver the preferred sense of depth for comfortable viewing. State-of-the-art techniques for displaying these images already exist but there is a problem of how to generate these multiple images. The generation process for these images is not a trivial task and still has problems. Two solutions have been presented to perform this task:

1. *Multi camera system:*

This begins by acquiring multiview images of the same scene from multiple cameras from different viewpoints as shown in Figure 1.2, but using many cameras has

disadvantages and is considered an expensive system. This system cannot produce continuous viewpoints; for example, if we need an image from a position between camera 2 and camera 3, this system cannot generate this image. Using many cameras also requires a lot of effort to calibrate the cameras and produces a large amount of data that are not effective for transmission. An alternative solution for this system is to use *"Intermediate View Reconstruction"*, which is explained below.

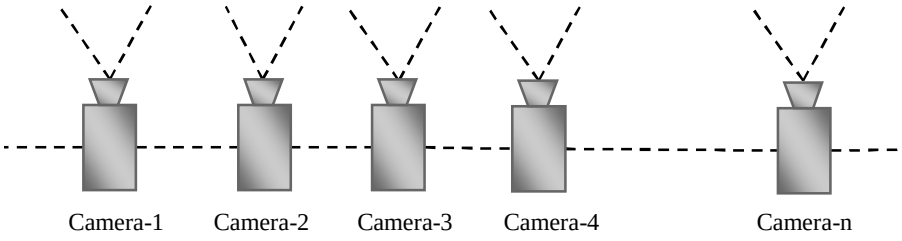


Figure 1.2: Multi camera system

2. Intermediate view reconstruction:

This approach involves generating additional images from the existing few images by positioning virtual cameras between a few real cameras to create virtual views at that position, as shown in Figure 1.3, where the two original real cameras are denoted by "L" and "R" and the other dotted cameras are the virtual ones denoted by V1, V2, etc. This approach is also called multiview image interpolation or view synthesis or image-based rendering. This approach is considered a main goal of this thesis, as it produces continuous viewpoints that enable us to experience a natural look-around feeling and a comfortable view to simulate the reality of viewing.

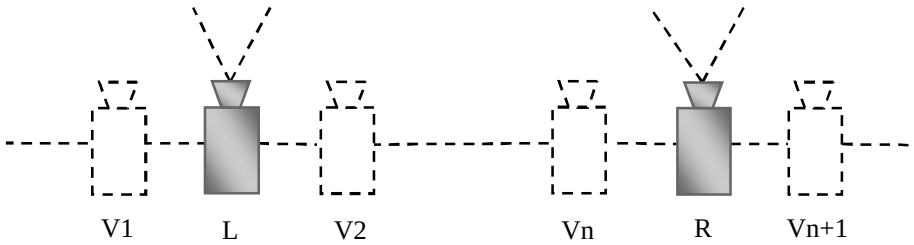


Figure 1.3: Original and virtual cameras in the intermediate view reconstruction process

Intermediate view reconstruction is a good approach to find additional images but there are problems in that the generation process struggles to give an accurate intermediate image. These problems present difficult challenges for the researchers. One of these problems is finding the disparity map, which has to be available before the reconstruction process because it determines the structure of the view. Since the image points in occluded areas are visible in one image, the disparity cannot be estimated there; thus, the occluded areas cannot be reconstructed in the intermediate image. Therefore, finding the location of the occluded areas is considered an important challenge for the reconstruction process. Evaluating the correct disparity in those areas is considered another challenge to the reconstruction of an accurate image.

In addition, there are a number of image artifices that are generated through formation of the intermediate view and that must be solved, such as the generation of holes and cracks due to image resampling, generation of ghost contours due to the disparity's sharp edges, and the appearance of disoccluded areas due to the cameras changing position. Moreover, because the intermediate image is generated from interpolation of more than one view, preserving the discontinuities in the reconstructed image is required and is considered a challenge to the reconstruction process. More details about these challenges are described in Chapter 3. Many researchers are working very hard to solve these problems, and a comprehensive review of their work is introduced in Chapter 3.

1.5 Application of Intermediate View Reconstruction (IVR)

Intermediate View Reconstruction (IVR) has many applications on 2D and 3D systems. In the following we will mention how it is applicable on each system separately.

1.5.1 Application of IVR on 3D System

Following recent advances in 3D display technologies, multiview displays have emerged and the need for multiple views has become an urgent requirement. Intermediate View Reconstruction (IVR) is a preferable solution to this because it generates the content of multiview displays using the minimum number of cameras.

In addition, it can be used to transfer the content of two-view stereoscopic displays to multiview displays using the same previous stereo image acquisition parameters. Also, it can be used to create additional images when the number of the available images is not the same as the required number of images for multiview display.

Through communication processes, IVR can be used to reduce the transmission bandwidth by sending a minimum number of images sufficient to generate additional images at the other end. Videoconferencing is an example of a communication process that uses IVR to make this process more efficient. Moreover, IVR provides the viewer with a natural look-around feeling with comfortable viewing. This is achieved by presenting multiple view windows, as depicted in Figure 1.4, and allowing the viewer to select the preferred sense of depth for comfortable viewing.

IVR can also be used to create the content of video games; for example, the Sony Company uses IVR for this purpose. Firstly, they use the IVR to create an intermediate image in a central position between the reference images. Then, they use the centre view and one of the reference images to create any additional images. This is done because, when using a smaller baseline, the interpolation errors will be minimized.

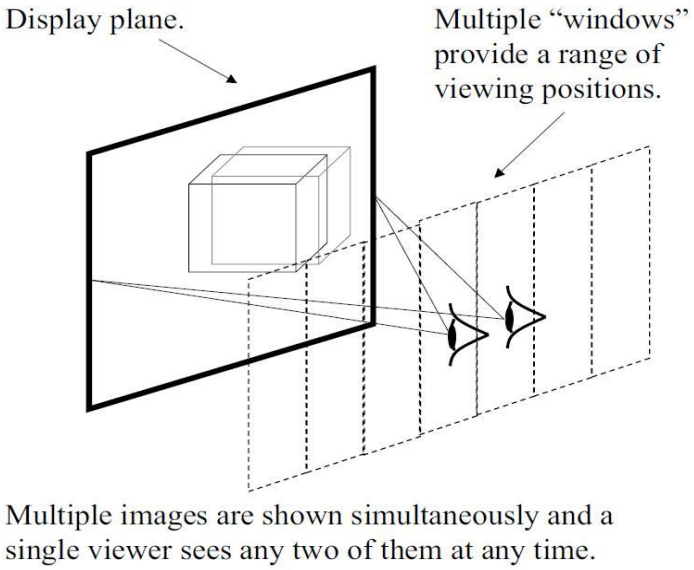


Figure 1.4: Demonstrating multiview display [5]

1.5.2 Application of IVR on 2D Systems

Following the recent advance in media technologies, the ordinary TV is becoming an unsatisfactory device when compared to the devices generated by the media technological revolution. Therefore, Free-viewpoint TV (FTV) has been developed to allow the viewer to see different views by interactively changing their position [16]. Displaying this wide range of viewpoints require the generation of multiple views from a limited number of available views [16]. FTV can be displayed on many different types of interfaces including 2D and 3D displays.

1.6 Our Contribution

The generation of the intermediate image from stereoscopic images is not straightforward, due to the ambiguity of the occluded region and the need to preserve the discontinuity of the edges. Therefore, many studies have been conducted to try to generate the intermediate images with high quality. The contribution of this research will be to develop a new algorithm to find intermediate images with high quality. Detection and handling occlusion problems are among the challenges of intermediate view reconstruction [17]. Therefore, it is necessary to develop efficient algorithms for detection and handling occlusion areas. Therefore, this thesis will offer the following main contributions:

1. Developing an efficient novel occlusion detection algorithm to detect the occluded/newly-exposed area based on the physical movements of the objects between the stereo images. Our algorithm has many advantages; for example, the occlusion area is detected without any fragmentation, the results are consistent under different types of images and matching algorithms, it needs few input parameters, and it is easy to implement. Qualitative and quantitative measures have confirmed that the proposed algorithm detects the occlusion areas with a high accuracy and lower error.
2. Developing an accurate block-based disparity inpainting algorithm for occlusion area recovery. This novel technique has been specially developed for occlusion area-filling in disparity images. Our algorithm has many advantages; which are, the occluded area is filled without any blurriness, and

edge discontinuities are preserved with a continuation of curved edges. We performed different analyses and computer simulations which confirmed that the proposed technique inpaints the occlusion area with a high accuracy and lower error.

3. These proposed algorithms are employed in a novel occlusion-aware intermediate view reconstruction. This novel algorithm focuses on how to find an intermediate image with a given disparity between two input images. An improvement in the backward-projection of the intermediate view is proposed by adding occlusion awareness to the reconstruction algorithm and proposing three quality improvement techniques to reduce image artifices as follows:

   3.1. Filling the re-sampling holes: instead of using a median filter to fill cracks and holes produced by image re-sampling, we proposed an object-based filling technique that fills the hole located in a foreground object or around its edges by an object's intensity. Subjective and objective evaluations confirmed that the proposed technique produces more accurate results around object boundaries.

   3.2. Removing ghost contours from the virtual view and restoring them to their correct place by extending the newly-exposed area three pixels in the direction where the ghost contour is located.

   3.3. Handling the disocclusion area: the disocclusion areas are recovered from one of the inpainted disparities based on the calculated visibility map.

   After that, the texture of the intermediate image is reconstructed either from the left or right images based on a robust and efficient selective strategy. This selection is performed upon the visibility of each image point in the intermediate image which is determined by a novel approach. Specifically, this approach is developed to detect the occluded/ newly-exposed area. Depending on this information, the visibility of the image points in the intermediate image will be determined.

4. Conducting subjective human trials to assess the quality of the reconstructed intermediate image on a 3D display relative to the reference intermediate image based on the ITU-R 500 recommendation and comparing it to the reconstructed images by backward and forward algorithms.

1.7 Thesis Outline

Here is a brief description of the contents of this thesis:

- **Chapter 2** introduces the main concepts of a 3D system and explores the main characteristics of the human vision system and how it is simulated in computer vision. This chapter also describes in detail the depth perception concept in three forms, the development of 3D display technologies, the geometry of the stereo vision, and the disparity estimation problem. The stereo vision geometry describes image acquisition through a pinhole camera to illustrate the relationship between two stereo images through epipolar geometry. This is followed by a description of how this epipolar geometry is exploited in the disparity estimation problem. Classification of the disparity estimation techniques is presented at the end of this chapter.
- **Chapter 3** presents previous studies on the intermediate view reconstruction problems. This chapter describes the challenges of intermediate view reconstruction and some of their intended solutions. A comprehensive survey on intermediate view reconstruction is presented on the classification of other studies and on our classification, which depends on how the disparity is compensated into the intermediate image. Moreover, a review of all the previous studies on intermediate view reconstruction is presented in a summarized table.
- **Chapter 4** focuses on the proposed occlusion detection algorithm which is considered one of the major challenges for intermediate view reconstruction. A detailed background of the prior work on occlusion detection is given. In this chapter, the proposed occlusion algorithm is compared to three well-known algorithms quantitatively and qualitatively. Analysis and computer simulation of the conducted comparison is presented in detail.

- **Chapter 5** describes the details of design and analysis of the proposed inpainting algorithms for occlusion areas that were detected in the previous chapter. Previous works on this area are introduced in this chapter. A comparison to the four well-known inpainting techniques is conducted. Several examples of inpainting occlusion areas are given.
- **Chapter 6** describes our novel framework on the intermediate view reconstruction that employs the two previous proposed algorithms on it. In this chapter, details of the problematic areas in the reconstruction process and their existing solutions are discussed. After that, the proposed framework on intermediate view reconstruction is explained step by step in detail. Meanwhile, illustrative examples of the proposed improvement techniques are presented through these steps.
- **Chapter 7** describes in detail a subjective human trial that was conducted to evaluate the quality of the reconstructed image in 3D display. Comparisons between the proposed reconstruction algorithm and the standard backward and forward projection approaches are conducted quantitatively using PSNR and qualitatively by a human trial test on 3D display. Different analyses of these comparisons are given in detail.
- **Chapter 8** provides a general discussion on the main contribution of this research to summarize the conclusions, followed by a list of open directions for future research issues.

CHAPTER

2

Main Concepts in 3D System

This chapter gives a general background to the main concepts of a 3D system. It begins by exploring the main characteristics of the human vision system and how it is simulated in computer vision. This chapter also describes in detail the depth perception concept in its three forms, the development of 3D display technologies, and the geometry of the stereo vision.

2.1 Human Visual System

Many efforts have been made in the field of computer vision to allow the computer to see the world as a human sees it, i.e. a three dimensional (3D) vision. Practically, the computer receives a two-dimensional (2D) image for a certain scene in the 3D world; in this image, most of the depth that exists in the scene is lost. This is because a human can see many viewpoints, whereas the computer receives just one viewpoint of the scene. When the computer captures one image of the world, the 3D scene of the real world is transformed to the 2D image and this transformation from 3D to 2D is called many-to-one [18]. On the other hand, if we want to reconstruct the 3D image from the 2D image, this transformation is very hard to accomplish because there are many possible ways of performing this transformation, which is called one-to-many [18]. This is related to the 'ill-posed problem' to which no unique solution exists.

Therefore, researchers have studied the characteristics of the human visual system to simulate it [18]. This simulation starts by using the raw data that are used by humans, and trying to imitate the human retina and brain processing to employ them in the computer vision to permit the viewer to see in 3D. Firstly, we will introduce the characteristics of the human visual system and then show how the computer

vision exploits these characteristics to see in 3D like a human. The human visual system characteristics are as follows:

- Humans have five senses, but vision is the most important as it provides us with much more information than the other senses.
- The human visual system doesn't require any effort to interpret a scene. The required interpretation is available within a tenth of a second [18].
- The human visual system uses a huge amount of databases relating to the real world. All of these data are stored by experience.
- The human visual system has two eyes, which are separated horizontally by an average of 65 mm [14]; in this way, the human sees two different views in each eye. Then the brain returns them as one view with the perception of depth. Thus, the human visual system is called "Stereopsis" or "Binocular Vision", which is mainly responsible for perceiving depth.

2.2 Depth Perception

Depth is defined as a relative distance from the viewer to a certain object in the real world, and depth perception is the ability to perceive this distance. This perception is performed by the human visual system, which consists of two eyes and the brain. Each eye receives a different view, which is projected onto the retina in 2D form; these views are in turn sent to the brain which processes them and returns them as one view with the desired depth. This depth lets us see the world in 3D.

But from where does the brain extract the depth? In fact, there are many forms of depth cues presented to the human visual system, including the following: binocular depth cues, which depend on the input from two eyes; monocular depth cues, which depend on the input from one eye; and oculomotor depth cues, which depend on the reaction of our eye muscles to the viewing object distance [19] [20].

2.2.1 Monocular Depth Cues

Monocular cues give us the ability to perceive depth with just one eye, which will be a pictorial image. The depth is extracted from the characteristics that exist in the 2D image, such as light and shadow, interposition of several objects, blurriness of some

objects, relative size of the same objects, and others. These cues are often used by artists to provide their works with a depth perception.

- Interposition.

  When an object is overlapped with another object they give us the ordering of depth or the relative distance to each other. The occluded object is considered a distant object, as illustrated in the Figure 2.1.

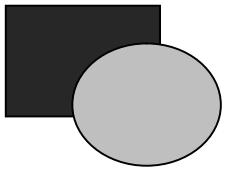


Figure 2.1: Interposition Depth Cue

- Light and shade

  This is considered a powerful cue to represent a depth in the 2D image, where the reflected light and their shadow give us information about the shapes of objects and their depth relationships, as illustrated in Figure 2.2.

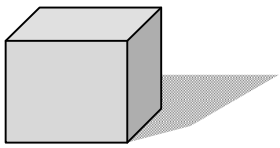


Figure 2.2: Light and shade depth cue

- Relative size

  Different sizes of the same object in the 2D image provide us with a cue to a depth existence, where the smaller object is considered further away than the larger one, as in Figure 2.3.

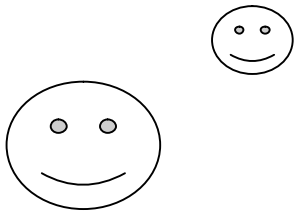


Figure 2.3: Relative size depth cue

- Aerial perspective

  Due to the existence of atmospheric phenomena, such as dust and fog, a distant object appears more blurred. Also, scattering light through the atmosphere affects the colour saturation of the distant object. This is illustrated in Figure 2.4.

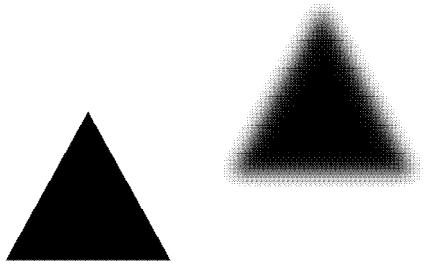


Figure 2.4: Aerial perspective depth cue

- Textual gradient

  A texture with uniform objects, such as a gravel road, gives us a depth cue in that the nearer objects are clear in shape and size but the distant ones become smaller, less detailed, and denser, as shown in Figure 2.5.

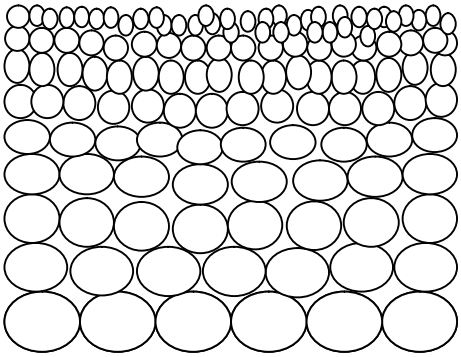


Figure 2.5: Texture gradient depth cue

- Linear perspective

  In fact, parallel lines appear to converge at the horizon line when they get further away from the viewer, and they thus provide us with a depth cue, as shown in Figure 2.6.

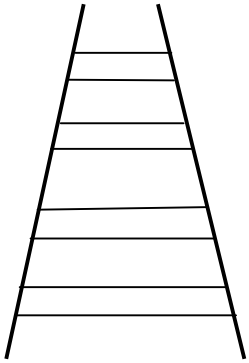


Figure 2.6: Linear perspective depth cue

- Motion parallax

  This cue is presented when either the object or the viewer's head moves. The object movements give us a depth cue as the distant objects appear to move more slowly than the nearer objects. This can be clearly seen when travelling by car.

2.2.2 Binocular Depth Cues

This depth cue is generated when each eye sees a separate 2D image of the same scene, and the brain then extracts the differences between these images to generate the perceived depth by combining the two images into one 3D image. This difference is called retinal disparity. If the object is far away the disparity will be small, while the disparity and the perceived depth will be large if the object is close. So, this binocular disparity provides us with information about the depth relationships between objects.

In the natural world, the human accommodates his/her eyes to focus on a certain point called a fixation point F. This point is projected into the same position in the retina, i.e. the retinal disparity is zero. Any other points in the scene that have a zero disparity as a fixation point F will lie on the same surface to form a *horopter* and are perceived as having the same depth of the fixation point F. The circular shape of the horopter is illustrated in Figure 2.7. On the other hand, any other points located in

front of or behind the fixation point are projected to different positions in the left and right retinas, forming retinal disparities that provide the viewer with a binocular depth cue.

Figure 2.7 demonstrates two examples of a retinal disparity; point A is located behind the fixation point F, so the difference between A and F will result in a positive disparity d'' in the retina and be perceived as a far point. Meanwhile, point B is located in front of the fixation point F and forms a negative disparity d' on the retina from the difference between F and B. Under natural viewing, the human can perceive a good depth within a restricted area around the horopter; this area is called *Panum's Fusion*. All objects located in this area are fused as a single image, while a double vision is perceived outside this area.

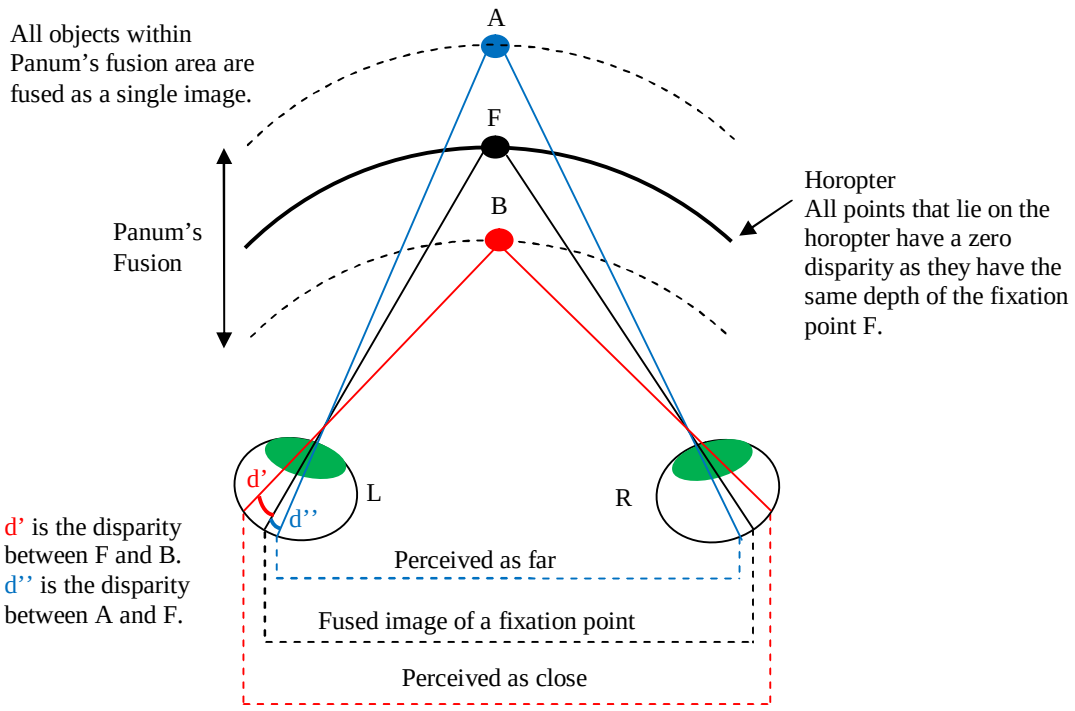


Figure 2.7: Natural binocular vision geometry

2.2.3 Oculomotor Depth Cues

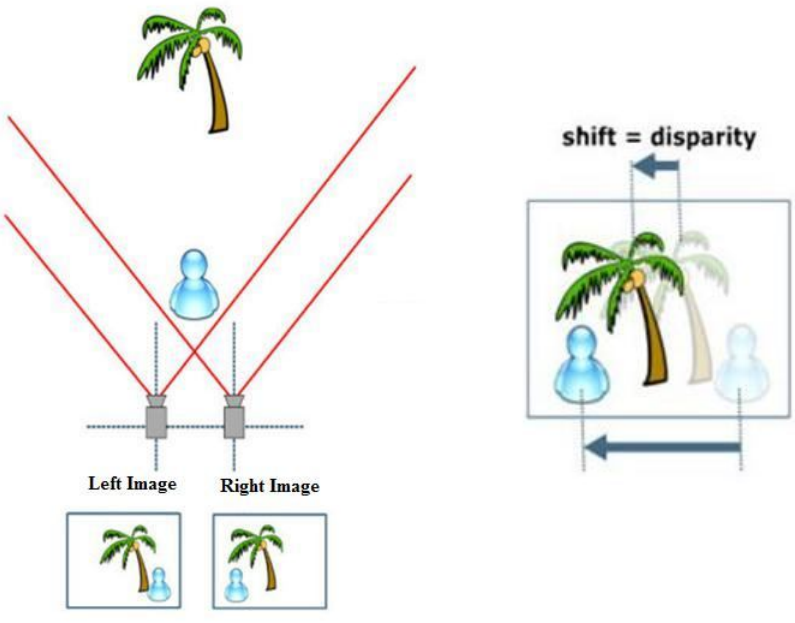
This depth comes from the movement of our eye lens's muscle. If the object is far away from the viewer, the lenses become thinner according to their focus on it; this is called the *Accommodation* of our eyes. Another Oculomotor depth cue is the

Convergence of our eyes which depends on the location of the focused object, where the angle of convergences will be smaller if the fixating point is far away from the viewer.

Monocular, binocular, and oculomotor depth cues work together to give more accurate depth cues than when each one works alone.

2.3 Stereopsis in Computer Vision

In computer vision, the human eyes are replaced by two cameras separated by the same distance between the human eyes and then acquire two different images; these two images are called *stereo images* and this system is called the *stereoscopic system*. Figure 2.8 (a) shows two different images (left and right stereo images) of a scene provided by different cameras. If we test these two images, we will see the differences between them which represent the *"disparity"*, as illustrated in the Figure 2.8 (b). This disparity can be defined as the distance between the same reference points of the two images. When these two images are seen by the viewer's eyes, each eye receives a corresponding image, and the viewer will perceive depth



(a) (b)

Figure 2.8: (a) Left and right stereo images (b) Difference between the left and right images (Disparity).

by analyzing them to generate a disparity that is similar to the retinal disparity of the real world. This can be achieved by using a stereoscopic display which directs the left and right images to the viewer's eyes to perceive depth. While the viewer's eyes are focusing on the display, any image points that have a depth will be displayed in front of or behind the display depending on the disparity between the corresponding points in the left and right images. Figure 2.9 illustrates the geometry of the perceived depth for two objects, one in front of the stereoscopic display and the other behind the display.

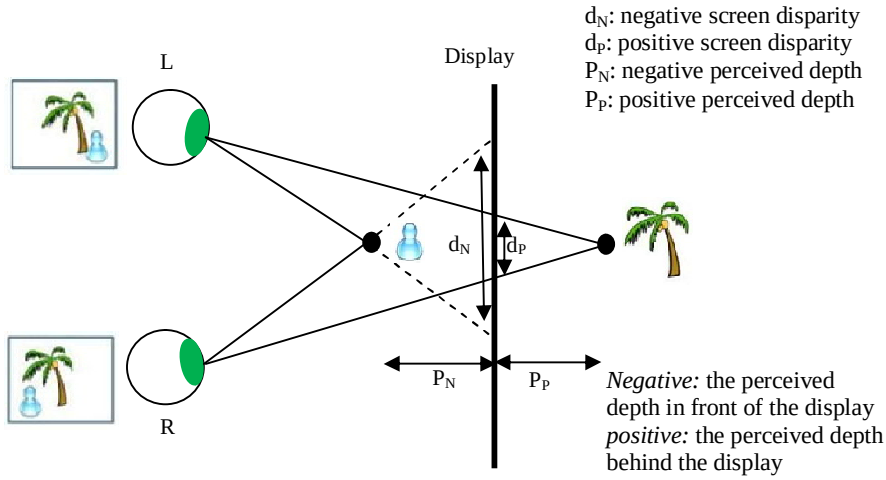


Figure 2.9: The geometry of the perceived depth for two objects in front of and behind the display

2.4 3D Display Technologies

Many types of electronic displays have been developed to simulate human binocular disparity which results from seeing two different images of one scene. Thus, the most important characteristic of the 3D electronic displays is the ability to give the correct image to each of the user's eyes to generate the perceived depth. These displays are categorized according to whether the viewer is wearing some kind of special device or not. Stereoscopic displays require the viewer to wear certain devices to direct the correct image to each eye; those devices may be eyewear or headsets. Autostereoscopic displays do not require the viewer to wear any devices to provide 3D perception, as the display directly provides the viewer's eyes with the

correct images. Specifically, a complete taxonomy of the 3D display is illustrated in the Figure 2.10.

As a point of fact, the left and right eye images cannot always be separated one hundred per cent by displaying them on a 3D display, but a percentage of one eye view will be visible in the other eye, generating *"crosstalk"*, which is considered uncomfortable for the viewer to experience. Most 3D displays suffer from this problem but the visibility degree of the crosstalk is different according to the technology that is used. Many studies have been conducted to determine the factors that affect the visibility of crosstalk, Pastoor [21] found that increasing the contrast and the disparity will increase the visibility of the crosstalk. Hence, the developer can alleviate the crosstalk visibility by avoiding these factors.

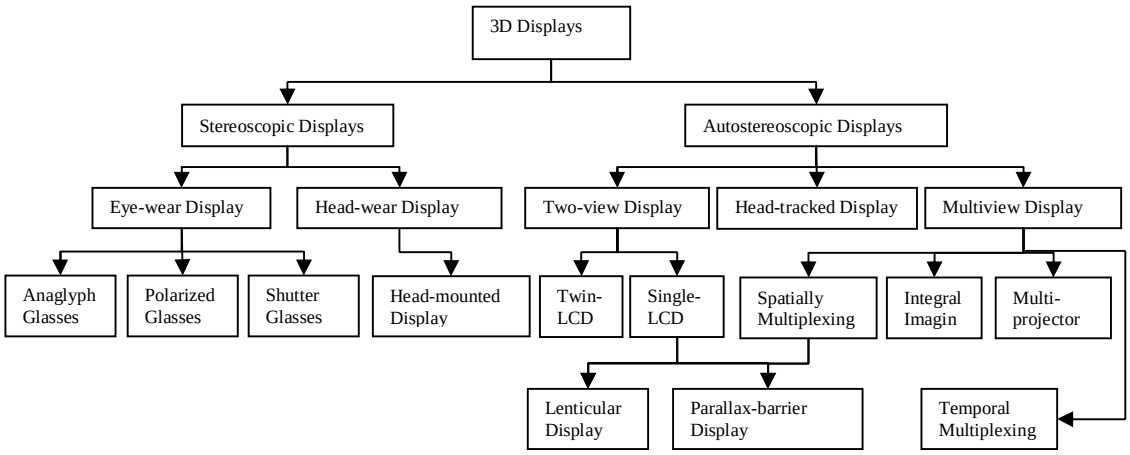


Figure 2.10: Taxonomy of the 3D displays

2.4.1 Stereoscopic Displays

Stereoscopic displays need to present the left image and the right image in the same display, where the viewer needs to wear special devices to separate the left and right images to the correct eyes. This type of display is considered a suitable system for multiple viewers such as in cinema, where providing each viewer with special glasses is easier than directing the left and right images to each person. These glasses are categorized according to the type of filters used, which are as follows:

a. Anaglyph glasses

These glasses have coloured lenses: red and blue for the left and right eyes respectively, which are considered filters. At the same time the anaglyph images are produced by superimposing the red component of the left image over the blue component of the right image. Because the glasses have the same colours, the red lens filters out the blue component and the blue lens filters out the red component. Since these glasses provide one image for each eye, the viewer perceives depth when wearing them. This method has its drawbacks, as it produces a high crosstalk and the colour that needs to be filtered out cannot be eliminated completely; thus, one of the eyes will see parts of the other eye's image, but the system is still used due to its simplicity and lower cost.

b. Polarized glasses

These glasses have polarized lenses which allow the rotated light with specific angle of polarization to pass through. So, the left and right images must be polarized at the same angles. This polarization is achieved either by using two projectors or by using polarizing micro optics. The former is done by using two projectors to display the left and right images, each of which must be sent to a certain filter to polarize each image with a specific angle and overlaid on top of each other. Polarized glasses allow only one image to pass to each eye.

When using polarizing micro-optics [22], there is no need to use projectors. The two images are spatially multiplexed on the same display, and the micro optics split it into two different polarized views which are directed to the correct eye when the viewer wears polarized glasses. This method is considered better than anaglyph glasses when comparing the crosstalk effect.

c. Shutter glasses

The left and right image are displayed alternately on the display; the lenses of these glasses are designated to be alternately closed and open in a synchronized manner with the suitable image on the display - when the right lens is closed the left image will be displayed and vice versa. This system is controlled by an infrared signal that is placed near to the monitor. The viewer does not notice these operations, which occur at very high speed.

d. Headset or head-mounted displays

Here, the viewer needs to wear a special device but not eyewear. The device consists of two small displays mounted on the head of the viewer with each display directed immediately to the viewer's eyes. Since the viewer sees two different images, the 3D image will be perceived. Recent technology has made these headsets light, but it is still not easy to isolate the viewer from the surrounding environment [23].

2.4.2 Autostereoscopic Displays

This type of display does not require the viewer to wear any device to see the 3D image, as it automatically separates the left and right images and provides them directly to the correct eyes [5]. Thus, the 3D display will appear a more natural kind of viewing because the viewer perceives the depth without needing to wear any device. This is accomplished by combining LCD display with micro-optic components which can be parallax barrier or lenticular sheet. This combination enables the display designer to develop various alternative technologies to autostereoscopic displays that can provide *two-view displays*, *head-tracked displays*, and *multiview displays*. For each of these technologies, there are many kinds of displays which are available theoretically and commercially. In the following sections we will mention briefly some of these display technologies and introduce their advantages and disadvantages, but we will concentrate mainly on multiview displays, which are related to our research area.

1) Two-view display

   The functionality of this display is to provide the viewer with two views, one for the left eye and the other for the right eye. This can be accomplished either by using two LCD displays, one for the left and the other for the right eye, or using a single LCD with two multiplexing images on it.

   Various technologies have been used to develop two-view twin-LCD displays [24, 25]. One of these technologies uses two LCD displays with a single light source [24]. Two mirrors are used to direct the light to each LCD display, each of which represents a different view, one for each eye, which are combined using a beam combiner. Since there are two displays for the viewer's eyes, he/she sees a full resolution of these displays. Another approach has been developed by Sharp

[25] using micro-optic twin-LCD displays with two light sources; behind each LCD display there are two arrays of optical elements that let the light pass to the LCDs in such a way that one LCD forms one view and the other one forms another view. This technology involving two LCD components is considered costly, especially for personal use, but it may be an acceptable cost for a big company.

Therefore, many technologies have been developed using different approaches to provide two-view display using a single LCD. The left and right images are spatially multiplexed on the same display where different kinds of optical components are used to direct each image to the correct eye. These optical components can be parallax barrier or lenticular sheet.

a. Parallax Barrier

   This was invented by F. Ives in 1903 [26]; it consists of strips of black mask placed on the front of the pixels to stop the light from passing to certain pixels. This allows the pixels to direct the light to two different views, half of them to the left view and the other half to the right view. Since the left and right images are interlaced on columns' pixels on the display, the viewer sees different columns in each eye. This technology provides the viewer with parallax in the horizontal direction only. Parallax barrier has drawbacks that affect the quality of the 3D image so many modifications of the LCD displays have been developed to overcome these drawbacks.

   Many detailed design questions about the parallax barrier have been analyzed in the literature [27, 28], such as using wider or narrower apertures, and whether it is better to place the parallax barrier behind or in front of the LCD element. Thus, using the correct design approach is very important since it provides a 3D image with a high quality and less crosstalk.

b. Lenticular Sheet design

   This was devised by a group of researchers in the 1930s [10, 12]; it consists of cylindrical lenses arranged vertically in front of pixels and is used to direct the light from every adjacent pixel to different views as the pixels are interlaced into columns on the display. Consequently, the viewer's eyes see

two different images in each eye. Also, the lenticular sheet technology provides the viewer with horizontal parallax.

Ocuity Ltd. developed a new design for LCD display using a switchable lenticular sheet [29]. This design comes from a novel architecture of Polarisation Activated Microlenses and consists of a combination of a passive birefringent material attached to a lenticular sheet and isotropic material. These Polarisation Activated Mirolenses are attached by a thin layer of switchable polarizer in front of them to allow the TFT-LCD display to work in 2D and 3D modes. In this way, a full resolution can be achieved in 2D mode, while a half resolution can be achieved in 3D mode. In addition, the 3D mode is characterized by the perception of lower crosstalk than when using an ordinary design of lenticular sheet, where it is <1.0% for a central viewing position.

Using two-view displays, the left and right images are visible in many viewing zones in the surrounding space; this means that the left and right views are repeated in the space many times. The viewing zone is defined as the correct place from which the viewer can see 3D image. These viewing zones are illustrated in Figure 2.11; they take the shape of diamonds where the viewer's eyes must remain within two viewing zones of them [5], and multiple viewers can hence see 3D images at the same time from different viewing zones.

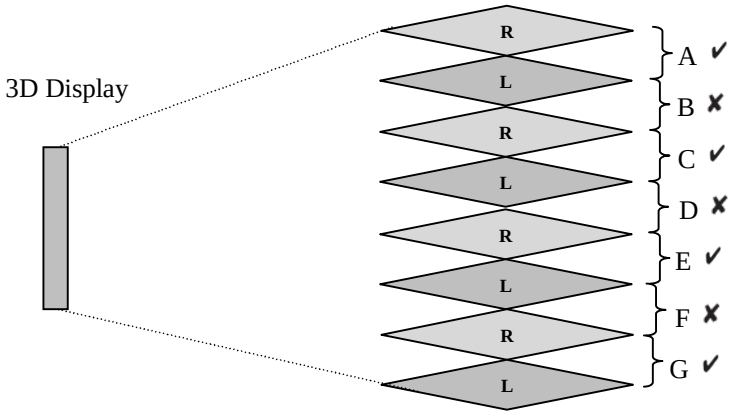


Figure 2.11: Correct and incorrect viewing zones

From the Figure 2.11 we can observe different viewing zones and each set of two viewing zones is represented by a letter. Each viewer's eyes must remain within the

correct viewing zones, with the left eye in the left eye viewing zone and the right eye within the right eye viewing zone. Thus, the correct viewing zones in the Figure 2.11 are A, C, E and G while B, D and F are the incorrect positions for the viewer's eyes. If the viewer's eyes are in the correct position, he/she will see the *orthoscopic image* (i.e. the left image is seen by the left eye and the right image is seen by the right eye), but if they are in the incorrect position, they will perceive a false depth effect and see an *psuedoscopic image* (the left image is seen by the right eye and the right image is seen by the left eye); consequently 50% of the psuedoscopic image will be seen. So, other technologies are required to solve this problem. One suggestion in regard to the two-view displays is to use Viewing Position Indicator (VPI) by [27, 30]; this Indicator helps the viewer to determine whether they are in the correct position or not. Although using VPI is helpful, it has some drawbacks. Therefore, the designers have sought another solution using other display technologies. In the following section we will introduce head-tracking displays and multiview displays which can solve this problem.

2) Head-tracked display

   This is the same as two-view display technologies with the addition of a head-tracking feature which follows the viewer's head movement to display the correct images to the appropriate viewing zone. In Figure 2.11, if the viewer's eyes move from viewing zone A to B, without the head-tracking feature, the left image will be seen by the right eye's viewing zone and vice versa; but, using the head-tracking feature, the display will follow the viewer's eye and swap the viewing zone to the correct position. Other technologies have been developed to provide only two viewing zones and allowing them to be moved when the viewer's head moves. For example, Xenotech [31] use two-projector technology to present two-views to the viewer's eyes, and the two projectors are moved to follow the viewer's head movement.

   We mentioned in the previous section that Sharp cooperation developed a two-view twin-LCD display using a single light source to give two viewing zones. This is enhanced by moving the light source to follow the viewer's head position and consequently moving the viewing zone. Sharp also developed a micro-optic

twin-LCD display which uses two light sources and two optical arrays for each display to follow the viewer's head movement by moving the two optical arrays simultaneously.

Although head-tracking technology is considered a good solution to the problem of the viewer seeing psuedoscopic images, there are many difficulties with tracking systems. The head-tracking operation must be fast and accurate without any noticeable delay. Moreover, the head-tracking system is only useful for a single viewer and, if the viewer's head moves, he/she will still see the same scene from the same viewing point, which does not occur in the real world. Therefore, new technologies are being adopted to provide more natural viewing by multiple viewers; these are introduced in the next section.

3) Multiview display

Multiview displays have been developed to overcome the problems of limited viewing freedom, which is a major problem in two-view and head-tracked displays. Since the multiview display provides multiple views at one time from different viewpoints, the viewer can see the 3D image from any viewing zone in the surrounding space, as well as seeing different images if he/she moves his/her head. Also, multiple viewers can see 3D images at the same time, with each one seeing different images from different positions.

Many technologies are being used to develop multiview displays; one of them, which divides the display resolution between multiple views, is called spatial multiplexing. There are other approaches such as using many projectors, each one for a single view, using a single very fast display to present multiple views sequentially, or a hybrid system which consists of a combination of two technologies. In the following section we will describe some of the work that has been done on these approaches.

a. Spatial multiplexing design

This principle is similar to the two-view displays using parallax barriers or lenticular sheets but, rather than being split into two views, it is divided into more views. The number of views depends on the pixel size and the resolution of the display [23], and also depends on the way in which the optical elements are

arranged. Consequently, just four views are visible on the LCD display due to the constraints on the pixel size and the resolution. To display four views for the viewers, four images have to be multiplexed horizontally in an appropriate way.

It is considered impractical to use parallax barrier in a multiview display because it blocks the light more than would be the case in two-views and, also, the barriers will be more visible as the number of views is increased [23]; it is more practical to use a lenticular sheet with LCD display to establish a multiview display. With a lenticular sheet, if the viewer's eyes move from one view to another, they will notice a dark line at the boundary because the lenticular elements enlarge the LCD pixels, producing a black mask between them [23]. Also, the resolution will decrease as the number of views increases. Philips [32] developed a new approach to overcome these problems by changing the arrangement of the lenticular elements and placing them at an angle to the LCD pixel array rather than arranging them vertically. With this new design, the images can be multiplexed horizontally and vertically, allowing seven views to be displayed with more reasonable resolution. Thus, these views will be crossed with one another, causing the reduction of the black line to be visible between views. With this solution, Stereographics [33] developed a display using the same technology with nine views.

b. Integral Imaging design

This technology was invented early in 1908 by Lippmann and proposed in [34]. Since the parallax barrier and the lenticular sheets give only the horizontal parallax, the integral imaging provides the viewer with full parallax (i.e. horizontal and vertical parallax). Lippmann discovered this by using an array of spherical convex lenses that are arranged horizontally and vertically; under each lens there is a complete 2D elemental image for the object from different perspectives. Hence, the viewer's eyes will see a 3D image when they move horizontally and vertically. This technology used the term "integral" due to the reconstruction of 3D image from the integration of all elemental images through the lens array. The integral lens sheet is placed in front of the LCD as in [35], and this LCD must be a high-resolution display because under each lens there is a complete image. Actually, using the current highest-resolution LCD displays

will not give a good-quality 3D image; for example, Okano et al. [36] used the integral lens sheet and a high-resolution LCD of (1280×1024), with resulting elemental images of (62×55) pixel due to the available LCD resolution. Thus far, no commercial display has been designed, as the required high-resolution display does not exist.

c. Multiprojector design

To generate a multiview display with a good resolution, multiple projectors are used, with one projector for each view and a single reflective screen to display all the projected views. The multiprojector approach must be accompanied by optical components which are joined to the display to direct the multiple views (i.e. projected images) automatically. The first multiprojector display using a lenticular screen was invented by H. Ives in 1931; he used 39 projectors and a single reflective screen, which was a lenticular sheet coated with a diffuser surface from the back to project all the views on it [37].

Many approaches have been developed in the literature on multiprojectors using a lenticular sheet [10]. One of the more recent approaches to a multiprojector lenticular display was developed by MERL [12]; they used two different arrangements for the multiprojector array and the lenticular sheet: *a rear-projection* and *a front-projection* display. In a rear projection approach, a double lenticular screen is used (i.e. two lenticular sheets are attached back-to-back with a diffuser surface between them), and the projectors and the viewers are placed on different sides of the screen. On the other hand, the front-projection approach uses only a single lenticular sheet with retro-reflective screen on the back, and the viewers and the projectors array are both in front of the screen. Figure 2.12 illustrates both approaches.

Multiprojectors display is considered costly as it uses a separate projector for each view and it is also very hard to align all of the projected images accurately above each other; however, when a good resolution is demanded, it is considered a good approach.

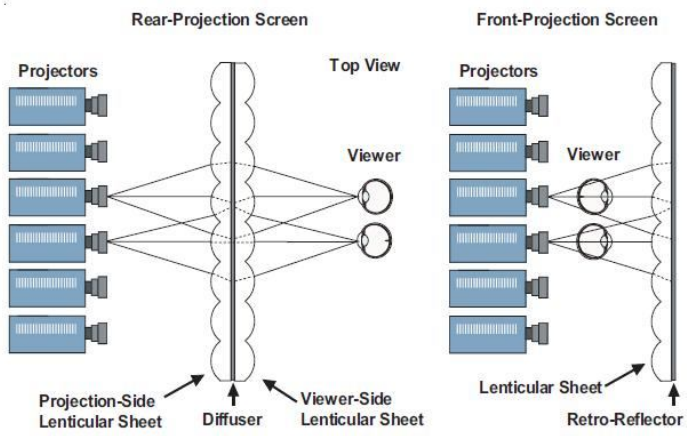


Figure 2.12: MERL multiprojector Display (rear- and front-projection display) [12].

d. Time-sequential design (Temporal multiplexing)

Time-sequential design is similar to the principle of shutter glasses in which the images are displayed sequentially on a single display at a high frame rate. Theoretically, the time-sequential multiview display is illustrated by using illumination bars; one of these bars is turned on to illuminate the high-speed screen which displays the images through the lens that is used to direct the displayed image to a certain viewing zone in the space [23]. In this way, the eye sees just one image at a time in the lit zone while the other zones are unlit (i.e. dark zones). So, multiview displays are generated each time one of the illumination bars is turned on; different images are displayed on the screen and this must be done at a high speed.

This remained theoretical until Cambridge University developed a practical display using this approach [38, 39]. Due to the requirement for a high-speed screen to effect rapid changes for images and the fact that LCD's speed is not sufficient for that, Cambridge used a high-speed CRT

e. Hybrid design

A combination of two techniques is used to increase the number of viewing zones, which was applied practically. One of them, combining the spatial multiplexing with multiprojector techniques to increase the number of views was used in [40] and [41] to produce 40 views and 72 views respectively. Also, a combination of time-sequential and multiprojector techniques was used by

Cambridge to produce 28 views with 25-inch display [38], and 15 views with 50-inch display [39].

f. Viewer-tracking multiview display based on LED scanning back-light
Since LCD display uses a full-luminance backlight to illuminate the monitor without taking into consideration the displayed image, unnecessary power consumption and low contrast ratio is gained. LED technology is developed to overcome LCD deficiencies by controlling the brightness of each individual block of pixels [], consequently, the power consumption is reduced and a high contrast ratio is achieved. A view-tracking multiview display with 8 views is developed using synchro-signal Light-emitting Diode (LED) backlight technology. This display sends different stereo image pairs depending on the viewer's position, in addition that can be watched by multiple viewers. The intelligent design of the dynamic backlight system had a low crosstalk and the displayed 3D image has a full resolution, thus a high image qualitatively is achieved.

2.5 Geometry of Stereo Vision

Stereo vision consists of two views, which are acquired either simultaneously via two cameras or sequentially by one camera moved slightly in relation to the same scene; although the former is more accurate, geometrically there is no difference between them. The use of a single camera transforms the 3D scene in the physical world to a 2D image on its imager, and this transformation is called *"projective transformation"*. Because both of the cameras project 2D images of the same 3D scene on their projection planes, the image points of the two images are related to each other geometrically. This relationship is derived from the basic concepts model of each camera which is known as a pinhole camera model. So, it is worth introducing the simple camera model and then illustrating the geometrical relationship between the stereo cameras.

2.5.1 Pinhole Camera Model

This is a simple and useful model to illustrate how the scene in the physical world is projected onto the image plane of the camera through a pinhole. A pinhole can be

defined as a tiny hole in the centre of the imaginary wall which allows the ray to pass just through this tiny hole [42]. Unfortunately, the pinhole alone is not able to collect enough light; therefore our eyes and cameras use the lens to collect more light, even though using a lens causes a small amount of distortion on the projected image due to either the shape of the lens itself or to inaccurate alignment. The former distortion is called Radial distortion and the latter is called Tangential distortion. Thus, the calibrated images need to undergo an undistortion process.

The projective transformation of a 3D scene of the world to a 2D image can be described by a pinhole camera model. Each ray reflected from the object in the real world and passed through a camera's pinhole represents a single point in the 2D projected image. The geometry of a pinhole camera consists of a *camera plane* with a pinhole aperture to allow the rays to pass through it, and an image plane (also known as a *projective plane*) which projects the rays as 2D points forming a 2D image on it. The distance from the pinhole aperture to the image plane is known as a *focal length* and that from a pinhole aperture to the object is referred as *Z*, as illustrated in the Figure 2.13. The point on the camera plane is called the centre of projection or the focal point *O*, the point on the image plane is called the principal point *C*, and the ray that passes through the centre of projection *O* to the principal point *C* is called the principal ray or the optical axis. The relationship between the point in the physical world and the point in the image plane is derived from the similar triangle in Figure 2.13 as following:

$$\frac{-x}{f} = \frac{X}{Z} \tag{2.1}$$

$$-x = f\frac{X}{Z} \tag{2.2}$$

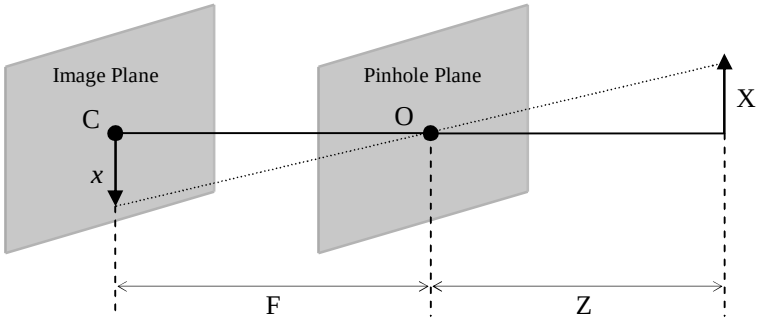


Figure 2.13: Pinhole Camera Model

The minus sign indicates that the projected image on the projective plane has been rotated; this occurs in real cameras due to the placing of the image plane behind the camera plane. To simplify this model geometrically, the image plane is placed virtually in front of the camera plane, as illustrated in Figure 2.14. Thus, the relationship becomes:

$$x = f\frac{X}{Z} \tag{2.3}$$

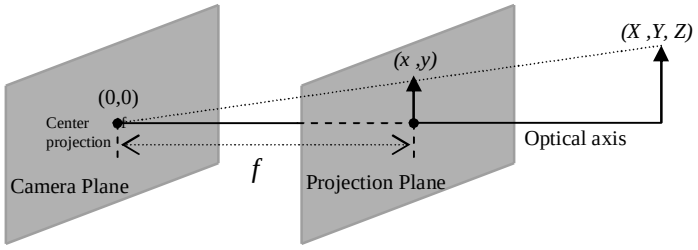


Figure 2.14: Simplified pinhole camera model

In fact, one can hardly align the image plane with the lens of a pinhole, so the principal point *C(0, 0)* is rarely aligned with the centre of the image. Therefore, new parameters for the principal points are introduced, $C(c_x,c_y)$, which refer to the displacement of the centre of the projection *C(0,0)* from the centre of the image plane. According to this displacement, the relationship above in equation (2.3) is changed to become that of equation (2.4), and the y coordinate will also be changed as in equation (2.5). This displacement is illustrated in Figure 2.15. On the other hand, because the individual image unit in the image plane is rectangular instead of

being square, the focal length will be expressed by two parameters, f_x and f_y. These parameters represent the result of multiplying the actual focal length F by the size of the individual image unit (s_x,s_y) as illustrated in Figure 2.15. Practically, f_x and f_y are not calculated; they are given.

$$x' = f_x\left(\frac{X}{Z}\right) + c_x \tag{2.4}$$

$$y' = f_y\left(\frac{X}{Z}\right) + c_y \tag{2.5}$$

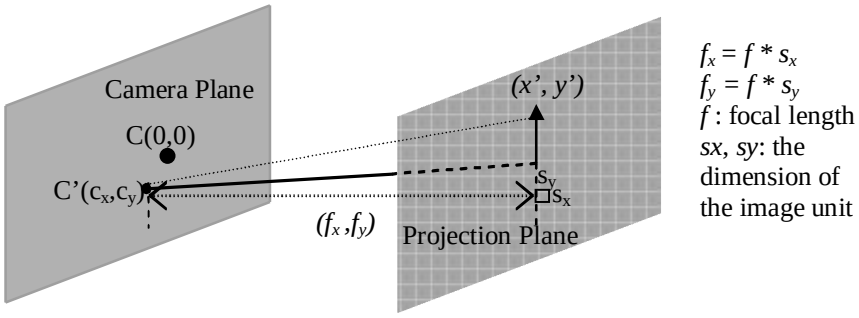


Figure 2.15: Misalignment of the image plane with the centre of the projection

Since the point in the physical world is represented by three coordinates *(X, Y, Z)* and its projective transformation on the projective plane is represented by two coordinates *(x, y)*, it is convenient to convert these coordinates to homogenous coordinates[1][43]. For example, we can convert the image point *(x, y)* to a homogenous coordinate by adding a third dimension to the point to become *(x, y, w)*. Because all the proportional points are equivalent, we can recover the original image point by dividing it over *w*. This allows the camera's parameters (f_x, f_y, , c_x, and c_y) to be arranged on a single (3×3) matrix. Thus, the transformation process becomes a multiplication process of a physical 3D point with a camera matrix to obtain the 2D image point, as shown below:

[1] The homogenous coordinate is the convenient way to express the coordinate while working in the projective transformation of 3D scene into a 2D image plane. Specifically, any point in the projective space with a dimension n in the homogenous coordinate is expressed with a dimension (n+1) vector. Also, when using those coordinates, any two proportional vectors are considered equivalent.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.6}$$

Obviously, from the result of multiplication process, the value of w will be equal to the value of *Z* and, since the image point is a homogenous coordinate, the original image point is a homogenous coordinate, and the original image point can be recovered by dividing by w or *Z* since *(w=Z)*.

Having introduced the basic model of a single camera (pinhole camera model), we now introduce the basic geometry of two cameras for a stereo imaging, explaining how they are related to each other geometrically. This relationship is known as epipolar geometry and will be illustrated in the next section.

2.5.2 Epipolar Geometry

This is the basic geometrical relationship of stereo images which are produced by two cameras projecting the same 3D scene in the physical world onto their imaging planes from different viewpoints. Since these two projected images represent the same 3D scene, their image points correspond to each other and the recovery process for these corresponding image points is called the stereo matching problem or disparity estimation. The question is: how can we recover the corresponding points from the two images? The answer to this question can be found through the epipolar geometry relationship of two images.

Firstly, we will mention the basic geometrical concepts about the cameras and the corresponding new epipolar geometrical terms, as illustrated in Figure 2.16. Each camera has its own focal point (i.e. centre of projection), O_l for the left camera and O_r for the right one. Further, each camera has its own projective plane (image plane) on which to project its image. The focal points of each camera, O_l and O_r , are projected onto each other's projective plane forming a new point on it; these are denoted as epipoles, E_l for the left projective plane and E_r for the right one. The point in the physical world is referred to as *X* and the corresponding image points are x_l and x_r. The back-projected rays from x_l and x_r intersect at X, and are thus lying on the same plane. Therefore, the points O_l, *X,* and O_r are forming a new plane referred

to as an epipolar plane π, and the intersection of this plane with the projective plane are forming lines called epipolar lines L_l (E_l-x_l) and L_r (E_r-x_r).

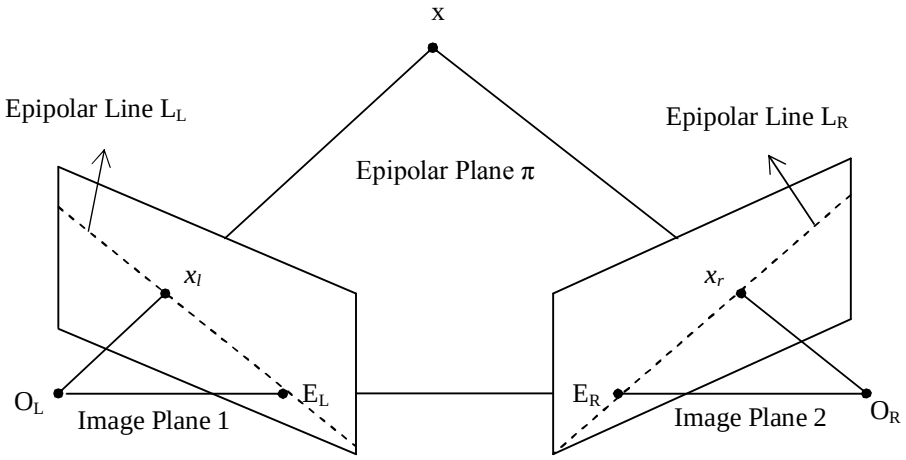


Figure 2.16: Epipolar Geometry

Accordingly, how can the epipolar geometry help us with the search for corresponding points on the stereo images? Since the point x_r is located on the epipolar line L_r and x_r lies on the epipolar plane π, so the epipolar line L_r is the image of the line *(O_l-X)* where x_l is located. Thus, the relationship of two corresponding points via epipolar geometry is concluded as follows: *all possible locations for one point in a projective plane are restricted to a certain line in another projective plane*. Consequently, epipolar geometry limits the range of the search for the corresponding image points between the stereo images, thus simplifying the stereo matching problem and increasing the reliability. The epipolar geometry is defined through a 3×3 matrix known as *Fundamental matrix (F)* that joins two points through this relationship: (x_l F x_r =0). The fundamental matrix is constructed from the camera matrices M_l and M_r which are introduced via a pinhole camera model.

2.5.3 Image Rectification

To further simplify the search process for corresponding points of a pair of stereo images, the imaging planes must be aligned exactly. Since, in reality, it is hard to align two cameras so that there are two exactly aligned image planes, an *Image Rectification* process is required to transfer the image planes of the two cameras to

the same common plane [44, 45]. So, after the rectification process, the epipolar lines will be parallel horizontally and parallel to the base line (O_l-O_r), and each image point and its corresponding point in the other image will be aligned horizontally via one row. Figure 2.17 shows a stereo camera setup after rectification process. Subsequently, the disparity between stereo images will occur only in the direction of the x-axis: ($d=x_l - x_r$). Subsequently, the depth can be extracted easily from the Figure 2.17 by triangle similarity:

$$\frac{B-(x_l-x_r)}{Z-f}=\frac{B}{Z} \tag{2.7}$$

$$Z=\frac{fB}{x_l-x_r} \tag{2.8}$$

Substituting ($d=x_l - x_r$) the equation becomes:

$$Z=\frac{fB}{d} \tag{2.9}$$

From the equation (2.9) we notice that, if the disparity is increased, the depth is decreased and vice versa. So, the depth is affected by the distance of the object; an object near to the camera will have a good depth.

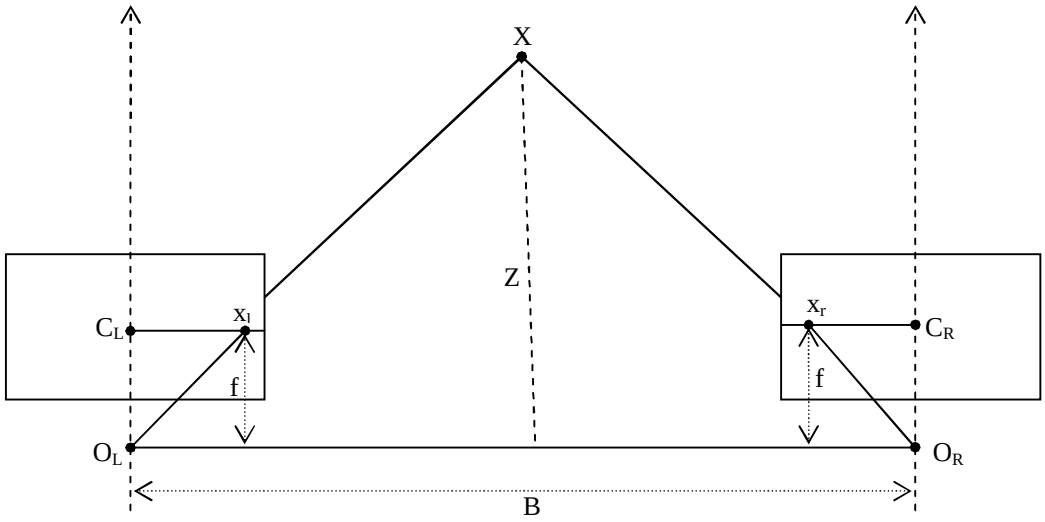


Figure 2.17: Rectified Image Planes

In conclusion, we will summarize the stereo imaging into the following steps [46], which are illustrated in Figure 2.18:

1. Remove the radial and tangential lens distortion from the acquired stereo images. This is achieved by rescaling and displacing the image points in the image plane using distortion coefficient to fix this distortion [47]. Consequently, an undistorted or corrected image is produced.
2. Rectify the stereo images to appear on the same common plane using epipolar geometry by aligning the epipolar lines horizontally and allowing the centre of the projection of the two stereo images to become parallel to each other [44]. By this, a row-aligned (having the same y-coordinate) or a rectified image is produced.
3. Match the corresponding points on the stereo image that refer to the same 3D points of the physical world to produce the disparity map, which represents the difference between the corresponding points in the x-axis coordinates as ($d=x_l - x_r$), since there is no difference in the y-axis.
4. Re-project the stereo image points to obtain the perceived depth by converting the disparity map to distances to produce a depth map.

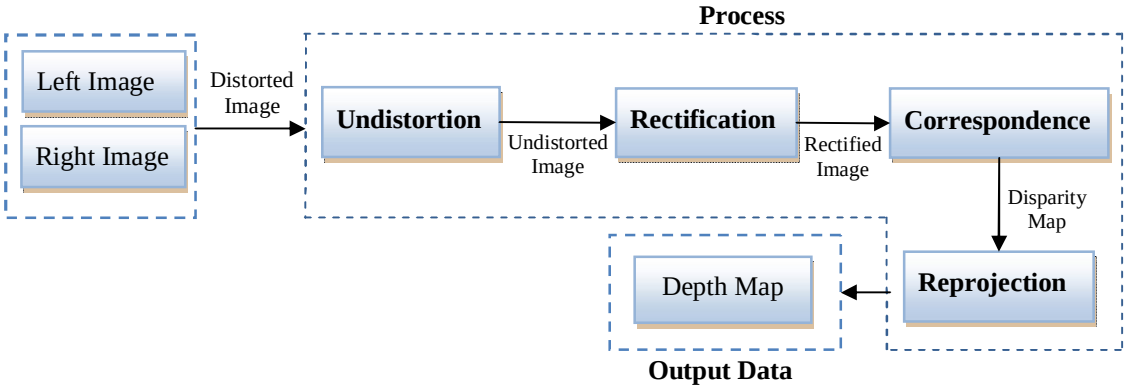


Figure 2.18: Stereo imaging steps

2.6 Disparity Estimation Problem

The stereo images result from projecting the same 3D scene in the physical world to two different image planes of two cameras. So, the 2D image points in the projective

planes are corresponding but in different places, because they are the result of the projection of the same 3D points. The problem is how to find the corresponding points in the stereo image pair. This problem is called the correspondence estimation problem or stereo matching problem. After the corresponding points are found, the distance between these points can be calculated; the distance calculation is called disparity estimation process.

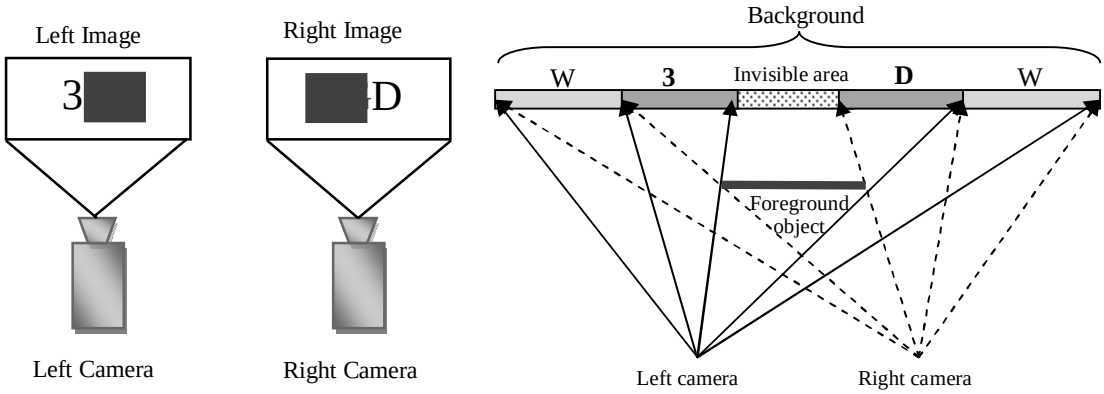
There are some constraints that help in the recovery process of the corresponding points in the stereo image pair. The first one is the Constant Image Brightness[2] (CIB) assumption [48], which assumes that the brightness of the image point does not change if it is shifted to a different place. So, the corresponding points have the same luminance. Although this appears a good constraint, it is not enough to solve the correspondence problems because, if there is an area (a group of pixels) in both images with the same brightness value, such as a contour, what is the corresponding point of each pixel in the other image?[49]. To solve this, the existence of other constraints is required.

The epipolar geometry relationship of the stereo image pair is considered the most important constraint as it limits the search process for the corresponding points just to one dimension where a point in one image lies on the epipolar line of the other image. Also, to further simplify the problem, the correspondence estimation problem is applied to the rectified images. Because these rectified images are on the same planes and aligned exactly, row-by-row, the correspondence search will occur only in the x-axis. So, when the corresponding points are found, the disparity can be calculated easily by calculating the horizontal separation between these points.

One of the factors that make the disparity estimation an ill-posed problem is the occlusion area, which means an area that is visible in one image but not in the other; for example, in the Figure 2.19, the world “3D” is half occluded in both left and right images due to the existence of a block in the foreground. In the left image the second

[2] Constant Image Brightness (CIB) assumption assumes that the corresponding image points between the stereo images have the same luminance value. In fact, this assumption is not always true since an object’s surface has reflection surfaces and this reflection value might be changed when acquired from a different position. Due to the difficulty of this situation, we will use this assumption and recommend work on the surface reflectance problem as a future research task.

letter “D” is occluded, whereas in the right image the first letter “3” is occluded. In this case, finding an accurate or exact disparity map is very difficult, thus making the correspondence estimation problem an ill-posed problem.



(a) (b)

Figure 2.19: Occlusion areas problem: **(a)** Placing the foreground object in different positions in left and right images; **(b)** Ordering the foreground object and the background in two layers to demonstrate the occlusion areas; “W” letter refer to the white background.

2.6.1 Disparity Estimation Techniques

Different techniques have been proposed in the literature to find an accurate disparity map using various approaches. In the following sections, we will discuss the main techniques used in disparity estimation.

2.6.1.1 Block-based techniques

In this technique, the stereo image pair is divided into rectangular blocks of pixels to match them up [50, 51]. The similarity between blocks is measured using one of the cost functions such as Sum of Absolute Difference (SAD). When the matching blocks are found, the disparity value is assigned to the all pixels in a block. This technique is desirable because it is simple, although it is computationally expensive. Block-matching techniques give good disparity estimation results in high-textured areas [49], but it is hard to estimate the disparity in low-textured areas and an additional interpolation step is required to estimate them.

Since the depth is changed abruptly at object boundaries and the block-matching technique assigns the same depth value for all pixels in the block, the result will have an incorrect value. Therefore, a block-matching algorithm with different block size is applied to overcome these problems [52]. Hence, the size of the block can be chosen adaptively depending on the applied area to strengthen the algorithm for low-textured areas and at object boundaries.

2.6.1.2 Features-based Techniques

In this technique, the matching process is carried out between certain features that are extracted from a pair of images [53, 54]. These features can be corners, edges, lines, or any feature that can be distinguished uniquely in one image and easily detected in the other image. Since the extracted features are unique, the feature-matching method gives accurate and reliable results but produces a sparse map. This means the disparity is computed only for the feature points and no disparity information is available for the other points. Although this technique needs less computation, it is considered inadequate for certain applications. For example, in intermediate view reconstruction algorithms, the disparity value is required for every pixel in the image to reconstruct all the pixels. In this case, the feature-based technique alone is insufficient but can be combined with another approach to provide a dense disparity map with more accurate results. In [55] a hybrid approach between a feature-based technique and a block-based technique is used to generate an accurate and dense disparity map. This hybrid method fails if the disparity values for two or more features are different and located in the same block.

2.6.1.3 Optical Flow Techniques

There is much similarity between the disparity estimation and motion estimation problems in computer vision applications. Both of them need to measure the displacement of pixels between two images. In both problems, the input is that two images have a correspondence between them, while the essential difference is in the time of acquiring images. In the stereo vision problem, the image pair is acquired at the same time but from different positions, while in the optical flow problem these images are acquired at different times. Therefore, motion estimation techniques can be exploited in the disparity estimation problem [48, 56].

There are some issues that must be taken into account when using the optical flow techniques in the disparity problem. In the disparity problem, the estimation is carried out on the rectified images, so the displacement distance will be in one dimension (x-axis); meanwhile the motion estimation problem has two components: horizontal and vertical. The difference between two images in motion estimation is very small and approximated numerically by computing a derivative of the intensity with respect to time, while the disparity range is larger. The displacement of a pixel can be measured and associated for every pixel in the image; this is called a dense optical flow or dense correspondence. As mentioned earlier, a dense map is required for certain applications, such as generating an intermediate image. Since the change in optical flow is very small, the optical flow algorithms apply an additional smoothness constraint to the whole image; therefore, the object boundaries will also be smoothed. In the real world, the object's surface appears to vary in smoothness with motion and preserves the discontinuity at the boundary, while optical flow techniques apply a smoothness constraint to the whole image, including the object boundaries. Many optical flow algorithms have been proposed in the literature [57, 58] that use an additional restriction in the optical flow equation to avoid smoothing across the object boundaries. The object boundaries can be determined explicitly or implicitly within the optical flow equation. Unfortunately, optical flow algorithms cannot detect a large motion (i.e. motion with more than one or two pixels) due to the numerical approximation of the derivative of the intensity with respect to time.

2.6.1.4 Phase-based Techniques

In this technique, the disparity estimation is extracted from the phase of Fourier Transform [59, 60]. The stereo vision exploits the fact that the spatial shift between the left and right view is linearly proportional to the local phase difference[3] between them, so the correspondence estimation is computed from that difference. The general idea of the phase-based method is to convolve the left and right images with a certain filter, such as a Gabor Filter, and then extract the local phase to compute the phase difference as well as the disparity.

[3] Local phase difference is defined as the difference between the band pass signals that extracted from the left and right images in the Fourier domain.

Since the difference can be computed for the all phase values, a dense disparity map will be generated. Moreover, the disparity map will be in sub-pixel accuracy without any explicit sub-pixel measurements. Also, the phase-based output is robust due to its stability even when the lighting and shading are variants. A comparative study between phase-based techniques for disparity estimation is found in [61].

2.6.1.5 Bayesian-based techniques

Since the correspondence problem does not have an exact match between all the image points in the left and right views (i.e. the ill-posed problem), a correspondence estimation based on a probabilistic approach has been developed [62, 63]. The Bayesian method is one of the probabilistic approaches that are used to find a simultaneous dense correspondence estimation. In the correspondence estimation problem, each point in the left image has many possible solutions in the right image. So, instead of selecting one of these solutions depending on a certain constraint, the Bayesian method assigns each possible solution a probability of being a solution or not, which are handled simultaneously. This assigned probability is extracted from two sources: the first one is the image content itself, and the second one is the prior knowledge about the image.

2.6.1.6 Energy-based techniques

In this technique, the disparity estimation problem is solved by a set of minimization and regularization formulations. These formulations are based either on statistical and discrete mathematical models or on variational and continuous models. The main objective here is to minimize the energy function cost which consists of the data and smoothness assumptions. Depending on the two mathematical models, the energy-based functions are divided into two types: statistical and variational approaches.

1. Statistical and discrete approach: in this type, the images are represented by one of the statistical energy functions, such as Markov Random Field (MRF), where the disparity is found by choosing the most probable match between the images. After regularizing the discrete energy function, the problem is to work out how to minimize this energy function. The minimization is often done by one of the

following algorithms: Dynamic Programming [64], Graph-cut [65], or Belief Propagation [66]. This statistical approach gives a good result with images that contain areas with a constant depth (i.e. by applying a depth constancy assumption); however, this approach may fail if the depth is varying smoothly such as in the presence of a curvature structure in the images. Moreover, the discrete presentation only allows integer values to be used in the disparity estimation process.

2. Variational and continuous approach: in this approach, the images are represented by a continuous energy function, and the disparity is found by minimizing this energy function. This minimization in variational approach is performed via a set of Partial Differential Equations (PDE) and there is a great deal of literature on this [67, 68]. PDE-based methods allow the representation of the images in a continuous surfaces, and that results in a grid-independence and isotropy [69]. As the continuous variational PDE-based methods represent the important geometric features of the images apparently, such as the gradient and curvature, this allows us to directly handle them [70]. It also allows us to simulate the dynamic processes easily, such as linear and non-linear diffusion [70]. Moreover, the PDE-based method can benefit from the huge amount of literature on the numerical analysis of PDE [70], which helps to yield an accurate disparity.

2.7 Summary

In this chapter, we introduced three main concepts about 3D systems: human depth perception and its simulation in computer vision; 3D display technologies; and the geometry of stereo vision. Firstly, we described the main characteristics of the human visual system and illustrated three forms of depth cues. Then we described how these characteristics are simulated in computer vision to develop 3D displays. We also demonstrated many of the 3D display technologies that have emerged since the first stereoscopic device was invented. Since the main goal of this thesis is to generate the contents of multiview displays, we presented more details on multiview 3D display technologies. After that, we described the geometry of stereo vision to

understand the relationship between the acquired stereo images from a pinhole camera through epipolar geometry.

At the end of this chapter, we illustrated how the epipolar geometry is exploited in the disparity estimation, and then described the main techniques used for disparity estimation from the simplest technique (block-matching) to the more complicated one (energy-based technique). We concluded that a sparse disparity map is not applicable for intermediate view reconstruction algorithms that need a disparity value for each image point in the image, *i.e.* dense disparity map. The feature-based technique is an example of a sparse disparity generator. Since the object surface in the real world is smooth, smoothness constraints are applied through disparity estimation. However, an extra awareness is required when applying such constraints on the image boundaries through intermediate view reconstruction.

Finally, we will not investigate the disparity estimation any further in this thesis since a substantial amount of work has already been done and is available in the literature. We will focus on the intermediate view reconstruction problem, which is reviewed in the next chapter.

CHAPTER

3

Challenges and Literature Review of Intermediate View Reconstruction

This chapter presents previous studies on the intermediate view reconstruction problems, describing the challenges of intermediate view reconstruction and some of the possible solutions. A comprehensive survey of intermediate view reconstruction is presented in regard to other studies' categories and to our category depending on how the disparity is compensated for in the intermediate image. Moreover, a review of all the previous studies on intermediate view reconstruction is presented in a summarized table.

3.1 Introduction

The process of generating intermediate images from stereoscopic images is not straightforward due to the ambiguity of the occluded regions and the need to preserve the discontinuity. Therefore, many studies have been conducted to try to generate the intermediate image with high quality. Thus, more detailed study of interpolation methods is required.

This chapter reviews other researchers' work on the intermediate view reconstruction. Other researchers' classification of the intermediate view reconstruction will also be presented. We will focus on algorithms designed to find and handle the occlusion areas, and those that are trying to reduce image artifices generated through formation of the intermediate view, all of which are considered difficult challenges.

3.2 Intermediate View Reconstruction Problem

There are three essential aspects that must be available in a stereoscopic system to achieve a sense of reality: depth perception, natural look-around feeling, and comfortable viewing. Unfortunately, the availability of just two images in a stereoscopic system provides us only with the perception of depth, which is not enough for real 3D viewing. Therefore, extending the stereoscopic system to a multiview system will solve this problem; this involves converting the available stereo image pair to multiple images. In a multiview system, a comfortable viewing sense is achieved by selecting a suitable image pair separated by the exact distance between the viewer's eyes; it also requires the natural look-around feeling that is provided when presenting a different pair of views to the viewer as the viewer's head moves.

State-of-the-art display technologies for multiple images of a multiview system already exist, but the process of generating these images is a hard task and still presents problems. The generation process is carried out either by using multiple cameras equal to the number of required images or by using a few cameras and positioning virtual cameras to create virtual views at that position. The former system is considered expensive while the latter is an efficient one and is called *intermediate view reconstruction.*

Intermediate view reconstruction is considered the main step to generating the content of a multiview display and is defined as generating additional virtual images from the available stereo images that are produced by a minimum number of real cameras. This is achieved by supposing that there are virtual cameras (V1, V2, ...) located anywhere between the real cameras or in the surrounding area to capture these virtual images, as shown in Figure 1.3. Intermediate view reconstruction is used to avoid the disadvantages of having to use many cameras, and to give a natural free viewpoint without any discomfort. Using many cameras cannot produce continuous viewpoints and also produces a huge amount of data that are not effective for transmission; however, using intermediate view reconstruction produces continuous viewpoints that provide us with a capability of "*motion parallax*".

Although intermediate view reconstruction is considered a good approach to generate multiple images, it is considered a difficult task due to the problems posed by the reconstruction process. These reconstruction problems include finding the disparity map, finding the location of the occluded areas, recovering the correct disparity in those occluded areas, preserving the discontinuity at edges, and forming the intermediate view texture without image artifices. Thus, solving these problems is considered a significant challenge for researchers.

In the subsequent sections we will explore the challenges of the intermediate view reconstruction process in more detail and review other researchers' solutions to these challenges. After that, we will classify the previous work on intermediate view reconstruction and mention the classification of other researchers.

3.2.1 Challenges of the Intermediate View Reconstruction

In the following section we will explain the most important challenges that will confront us in the reconstruction process. By solving these challenges, we may be able to generate an intermediate image of high quality.

The main challenges of intermediate view reconstruction are as follows:

1. *Disparity Estimation*

   Disparity estimation is considered one of the challenges that must be solved before finding the intermediate images because it determines the structure of the view. Since the disparity estimation is considered a matching process between corresponding pixels of two images, finding an accurate disparity is a difficult task due to the occluded areas between these images. Therefore, the disparity estimation is an ill-posed problem and many studies have been conducted to solve it. Previous solutions to the disparity estimation problem are summarized in chapter 2, section 6.

2. *Detection and handling of the occlusion areas*

   Occlusion areas have an impact on two problems: disparity estimation and intermediate view reconstruction. Since occluded areas are defined as the visibility of the image points in one image but not in the other one, and the

disparity estimation depends on the intensity matching, the correct disparity value will not be calculated in these areas. Consequently, an erroneous matching will be generated because the disparity estimation methods cause a disparity to be calculated, even in the occlusion areas, by choosing the best estimate between the stereo pair in the occluded areas. Finding where these occlusion areas are located is considered the first part of this challenge and this is known as detection of the occlusion areas. After finding the occlusion area, how will the correct disparity be calculated there? This is the second challenge that must also be solved and it is called *occlusion handling* or *disparity extrapolation or inpainting*.

Intermediate view reconstruction and occlusion area detection are related to each other in two areas. Firstly, since the quality of the intermediate image depends on the accuracy of the disparity estimation, and the occlusion areas struggle finding an accurate disparity map, the occlusion areas will affect the reconstruction process. The second issue is related to the visibility of image points in the intermediate image, where some points are visible in one of the images. Therefore, the reconstruction process must be aware of the visibility of points in order to reconstruct the correct information from the correct source. How to determine the visibility of points in the intermediate image is considered another challenge.

3. *Formation of the intermediate view texture*

After finding an accurate disparity and collecting information about the location of the occlusion areas, determining the intermediate view texture is the next step. Accordingly, how can the intermediate image texture be computed through the disparity map? Let us assume the distance between the left and right images is normalized to become the left image at position 0 and the right image at position 1, with the intermediate image at position α where α lies between 0 and 1. Under the CIB assumption [48], any intensity value x in the left image will be equal to the intensity value of the right image at $x - d_L(x)$, where $d_L(x)$ is the disparity value for x, summarized as follows: $I_L(x) = I_R\ (x - d(x))$. Therefore, the intermediate image intensities might be computed using the above relationship as follows:

$$I_{Int}(x - \alpha d_L(x)) = I_L(x), \quad OR$$
$$I_{Int}(x + (1-\alpha) d_R(x)) = I_R(x) \tag{3.1}$$

Unfortunately, the $\alpha d(x)$ distance may not yield an integer value and consequently will not belong to the sampling grid of the intermediate view, so irregular points will be generated. Converting these irregular points to regular points to form the texture of the intermediate image is considered a challenge [71]. In addition, there are a number of image artifices generated through formation of the intermediate view that must be solved, such as the generation of holes and cracks due to image resampling, generation of ghost contours due to the disparity's sharp edges, and the appearance of disoccluded areas due to the cameras changing position [72].

4. *Preserving the discontinuity of the edges in the intermediate image*

   Depending on the fact that the object has a smooth surface appearance in the real world, in estimating the texture of the scene (disparity map) one needs to apply a smoothness constraint to the objects in the scene. This smoothness constraint can be applied, for example, in the block-based matching method [50, 51] when assigning all the block points the same disparity value. Thus, if there is a boundary area within a block, this boundary will have to have a disparity similar to the non-boundary area, resulting in smooth edges. In energy-based methods [64-67], which are used frequently to estimate the disparity, each energy function has a smoothness term to be minimized and the neighbouring points in a small area will have to have the same disparity values. This constraint is applied to the whole image even in the boundary areas; consequently, the discontinuity will not be preserved.

   In addition, as a result of the reconstruction of the intermediate image, smooth edges will be generated since the intermediate image texture results from the interpolation of more than one view. Therefore, it is necessary to preserve the discontinuity in the reconstructed intermediate image and this is considered a challenge to the reconstruction process.

After summarizing the most important challenges confronting the reconstruction process, we will start to explain each one in more detail and review other researchers' solutions to these challenges.

3.3 Formation of the Intermediate View Texture

Since the intermediate view reconstruction generates additional images from the available few images, the unknown intermediate images will be derived from the available known images via their disparities. Initially, depending on the assumption of CIB [48], the left and the right images are related to each other via their disparities, as in the following relationships:

$$I_L(x) = I_R\big(x - d_L(x)\big) \qquad \forall x \in S_L \tag{3.2}$$

$$I_R(x) = I_L\big(x + d_R(x)\big) \qquad \forall x \in S_R \tag{3.3}$$

Where I_L is the left image and I_R is the right image that defined on the sampling grid S_L and S_R, respectively. $d_L(x)$ is the disparity map that is defined on the sampling grid of I_L toward I_R, while $d_R(x)$ is defined on I_R toward I_L. Now, let us introduce the two approaches which can be used to reconstruct the texture of the intermediate view depending on how the disparity values are calculated [71].

3.3.1 Backward Disparity Compensation

In this approach, the disparity values are defined on the sampling grid of the intermediate image $I_{int,}$ which will be reconstructed, toward both directions of I_L and I_R. Thus, the disparity map $d_{int}(x)$ will be defined on the sampling grid of the unknown images (i.e. the reconstructed images) to the known images (i.e. the left and right images). Therefore, each image point of the intermediate image will have a value using the intensities from I_L and/or I_R. Illustratively, let us assume the distance between the I_L and the I_R is a normalized distance, where I_L is located at 0 and I_R at 1 while the intermediate image is located at position α which must be within [0,1] (i.e. *if* $\alpha = 0$ then $I_{Int}=I_L$ while *if* $\alpha = 1$ then $I_{Int}=I_R$), as shown in Figure 3.1.

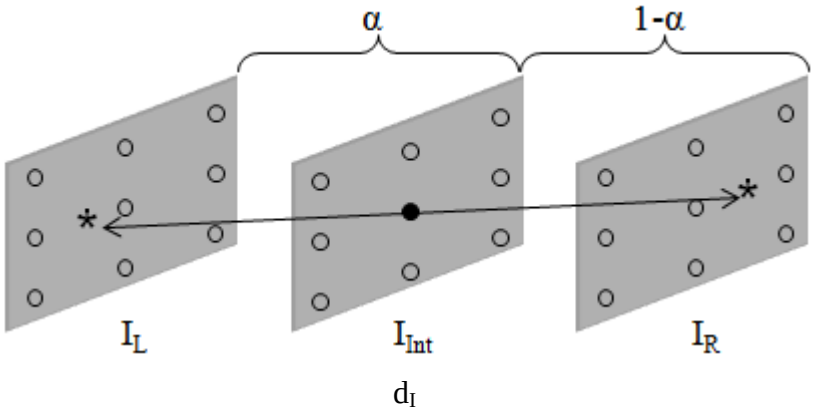


Figure 3.1: Shows backward-disparity estimation (disparity compensation from intermediate disparity to the left and right disparity)

Under the CIB assumption, the relationship between the I_L and I_R will be:

$$I_L(x + \alpha d_{Int}(x)) = I_R(x - (1-\alpha)d_{Int}(x)) \qquad \forall x \in S_{Int} \qquad (3.4)$$

Where $x + \alpha\, d_{Int}\,(x)$ and $x - (1 - \alpha)\, d_{Int}\,(x)$ may not necessarily belong to the sampling grid of S_L and S_R, respectively. However, this problem can be solved easily by using spatial interpolation. Then, the intensities of the intermediate image will be calculated using the linear interpolation (*i.e.* weighted averaging) of the intensities of the left and right images as in (3.5) [73, 74].

$$I_{Int}(x) = (1-\alpha)I_L(x + \alpha d_{Int}(x)) + \alpha I_R(x - (1-\alpha)d_{Int}(x)) \qquad \forall x \in S_{Int} \qquad (3.5)$$

There are other methods of reconstructing the intermediate image either from the left image or from the right image as in [75]. During the reconstruction method, the interpolation is applied twice, one when letting $x + \alpha\, d_{Int}\,(x)$ and $x - (1 - \alpha)\, d_{Int}\,(x)$ belong to S_L and S_R (spatial interpolation), respectively, and the other when computing the final intensities of the interpolated images. This will produce a blurred image which is considered a drawback of this approach.

Although the pivoting-based method produces a blurred image, Ince *et al.* [17] detected the edges in the intermediate image and found that the edge maps are very similar to the original intermediate image. Therefore, they used this reconstructed intermediate image as an input to the edge-preserved disparity regularization (i.e.

anisotropic regularization) to reconstruct a more accurate intermediate image without blurring at the edges.

3.3.2 Forward Disparity Compensation

In contrast to the backward disparity compensation, the forward disparity compensation approach computes the structure of the intermediate image via a disparity map that is defined on the sampling grid of one or both of the available images (left and right images), as shown in the Figure 3.2. Depending on the CIB assumption [48], the intermediate image intensity values are computed using the following relationships:

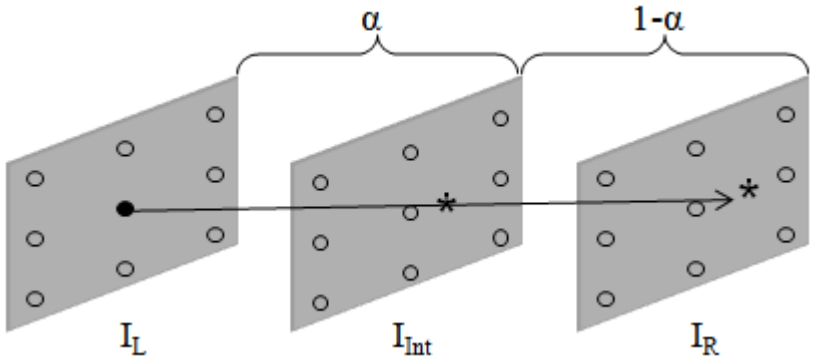
$$I_{Int}(x - \alpha d_L(x)) = I_L(x) \qquad \forall x \in S_L \tag{3.6}$$

$$I_{Int}(x + (1-\alpha) d_R(x)) = I_R(x) \qquad \forall x \in S_R \tag{3.7}$$

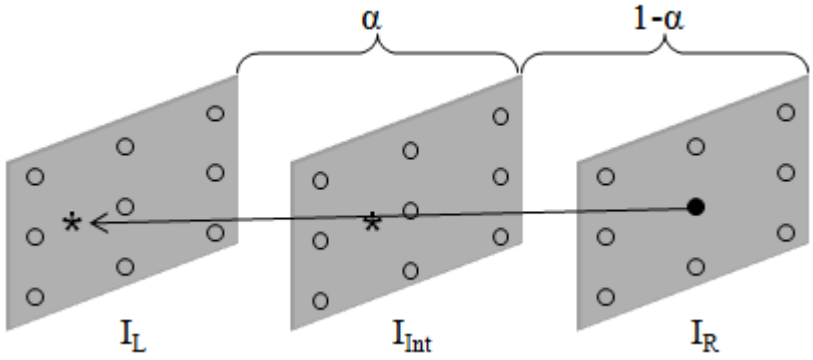
Unfortunately, $x - \alpha\, d_L(x)$ and $x + (1 - \alpha)\, d_R(x)$ locations will not belong to the sampling grid of the intermediate image S_{Int} (they are not integer values), resulting in irregular samples. Some researchers solve this problem by applying an additional step to force these irregular samples to belong to S , such as rounding these point locations to the nearest integer (i.e. $round(x - \alpha\, d_L(x))$) or applying a constraint where $\alpha\, d_L(x)$ and $(1 - \alpha)\, d_R(x)$ must belong to the S_L and S_R, respectively [76, 77].

Unfortunately, these solutions are not effective since some points in the intermediate image are not assigned intensity values (undefined points) or are assigned multiple intensities (overdefined points) as illustrated in Figure 3.3. For these areas, a texture synthesis can be used to handle the undefined points or a depth ordering can be used to choose a suitable intensity for the overdefined points [77]. Even though each point in the resulting image will have one intensity value, the image will be distorted. Thus, a method for converting the irregular samples to regularly-spaced intensities is required.

Recently, a spline-based reconstruction has been proposed to solve the problem of regularly-spaced samples in the reconstruction process [71]. This method is based on the minimization of the energy function which consists of a data-matching term for



(a) d_L



(b) d_R

Figure 3.2: Shows forward disparity estimation: (a) disparity compensation from left-to-right; (b) disparity compensation from right-to-left.

B-spline model of the irregular image samples and a B-spline smoothness term. Although this minimization gives a continuous function that recovers the regular sample space for the intensities, it needs a lot of computation, which is considered a drawback of this method.

The comparison between the spline-based reconstruction method and backward-projection methods is performed in the presence of image noise in one experiment and in the presence of error in the disparity estimation in the other one [71]. The former is more reliable for image noise, while, in case of an error in the disparity estimation, the latter one is more reliable.

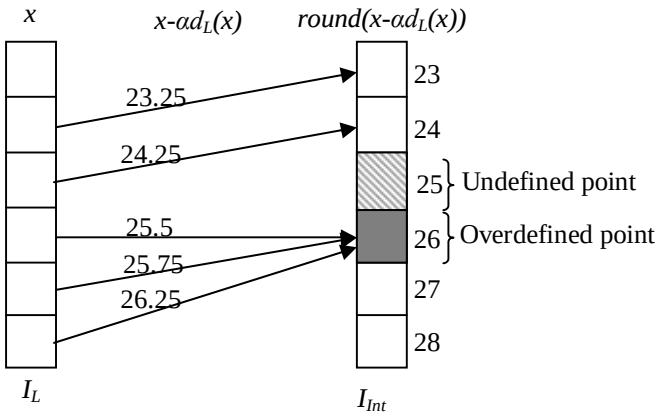


Figure 3.3: Example of the generated undefined and overdefined points through forward projection of image points from left image I_L to intermediate image I_{int}.

3.4 Detection and Handling of the Occlusion Areas

Since the stereo image pairs are acquired from different positions, they will have a different scene structure that results in the occluded area or newly-exposed area (uncovered). The occluded area is defined as the visibility of such an area in one image and its disappearance in the other one, whereas if the disappearing area becomes visible, this area will be called the newly-exposed area or uncovered area [17]. Figure 3.4 shows the original stereo pair I_L and I_R and the reconstructed intermediate image I_{int}. As we note, from left-to-right, the letter "C" is considered an occluded area because it is visible in I_L and has disappeared in I_R, while the area "T" is a newly-exposed area.

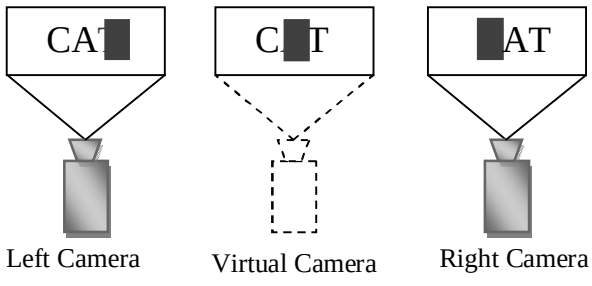


Figure 3.4: Occlusion problem through the reconstruction process

The occlusion problem is considered the major challenge in the reconstruction process. In order to reconstruct a correct and accurate intermediate image, there are

three sub-problems related to the occlusion area that must be solved. Firstly, it is necessary to discover where the occlusion area is, because the disparity estimation methods force the disparity to be calculated in the whole image, even in the occluded area. The second sub-problem is how to estimate the correct disparity in the occluded area that was detected. Finally, it is necessary to determine the visibility of the image points in the intermediate image to reconstruct the correct information from the correct source. As we can see in Figure 3.4, some image points I_{int} are visible only in I_L such as "C" and other points are visible only in I_R such as "T". Therefore, it is necessary to find a method to determine the visibility of image point in the intermediate image.

Here, a brief review of how previous work has detected and handled the occlusion areas will be discussed, as a detailed description of previous work on occlusion detection and handling are presented in the next chapter.

3.4.1 Methods and Constraints of the Occlusion Detection

Due to the importance of occlusion in many areas, many algorithms and constraints have been proposed to detect and handle those areas. Occlusion detection based on multiple images (more than 3 images) has been proposed in the literature [78], but this involves an extra computational effort. The simplest approach to detecting the occlusion area is the Left-Right Checking (LRC) approach [79-81], which uses the intensity matching between stereo pairs as an indicator of occlusion area. Ordering constraint (ORD) [82, 83] assumes that the order of pixels in the corresponding rows between the stereo images is not changed, unless an occlusion area is found. This constraint fails when there are thin foreground objects or narrow holes in the image.

LRC and ORD constraints use intensity matching, which is not robust in the presence of image noise, resulting in an erroneous occlusion area in the occlusion map. Occlusion constraint (OCC) [83, 84] detects the occlusion areas more accurately since it does not depend on the intensity matching but is based on the principle that the occlusion areas leave unmatched points in the other image, leaving an uncovered area near the boundary. Unfortunately, since this constraint is totally dependent on the disparity map, it fails with images that have a varying depth.

Another constraint is the uniqueness constraint [85], which checks one-to-one correspondence between stereo image pair. Because the disparity estimation methods apply boundary-preserving smoothness constraint [85], the gradient at the object boundaries is high; otherwise it is small [86]. Thus, the occlusion areas have a high gradient since they are located near to the boundaries; this constraint is called smoothness and gradient constraint. The visibility constraint [87] ensures that the visible pixels have at least one match on the other image, while the occluded area has no matches. This algorithm is flexible since it allows many-to-one matching but, sometimes, this is considered a weakness of this constraint.

Related to the visibility constraint, a geometric approach [88] is proposed to detect the occlusion/newly-exposed areas. This approach detects an empty area from projection in the target image after forward projecting the disparity map from reference to the target image. Recently, Phillips [89] have exploited the smoothness and gradient constraint of the occlusion area to detect it. They apply an adaptive boundary-preserving filter to highlight the occlusion area at the discontinuity of the calculated smoothed matching map. A comparison between five well-known algorithms is conducted in [90].

3.4.2 Methods and Constraints of the Occlusion Handling

After detecting the location of the occlusion areas, a robust recovering procedure for those areas is required. By considering the occluded areas as corrupted regions in the image, which need to be repaired, we can apply digital inpainting techniques to restore those areas which were originally implemented to remove unwanted objects from the image or to repair a corrupted image, i.e. image restoration. The simplest approach to filling the occluded area is horizontal interpolation using depth information (depth consistency assumption) that assumes the depth will be the same within small neighbouring areas [58, 76, 91]. A more complicated approach is to use Bertalmio *et al's* inpainting approach, which fills the gap by continuing the structure of the surrounding areas [92]. This approach is exploited in [93] to inpaint the occlusion areas in the disparity maps, where the filling procedure in the disparity is guided by stereo image gradient. In due course, several studies were conducted to improve Bertalmio *et al's* inpainting approach, as it produces a blurred filled region.

Therefore, exemplar-based inpainting techniques are proposed [94-96] to overcome this drawback; this method fills the gap by diffusing the structure of the surrounding areas, taking into consideration the texture replication to solve the blurriness problem. Modifying Criminisi *et al.*, a exemplar-based inpainting technique [94] is proposed in [97] to cope with occlusion filling.

A Laplacian filling technique (i.e ROIFILL MATLAB function) is used to fill the occlusion in [98]; this diffuses the surrounding area smoothly in the gap by solving the Laplacian equation. Unfortunately, an excessive smoothness is generated in the filled region, and unfilled holes remain. A more complicated technique uses the Field of Experts model [99] which consists of a group of learned filters; this draws on a database of real images with diversity to inpaint the occluded areas. This technique produces considerable results in combating blurriness but it is time-consuming. Other studies apply a smoothness filter such as averaging filter [100], symmetric Gaussian filter [100], and asymmetric Gaussian filter [101] to fill the occluded areas in the depth map. Unfortunately, this smoothness causes a geometrical disruption in a different direction depending on the filter used. A modified inpainting approach is used to fill the occluded area where it is filled from the background side, discarding the foreground side [102]. Comparisons of the inpainting techniques of occlusion areas are presented in [103, 104].

3.5 Intermediate View Reconstruction Algorithms

The generation of the intermediate image from stereoscopic images is not straightforward, due to the ambiguity of the occluded region and the need to preserve the discontinuity of the edges. Therefore, many studies have tried to generate intermediate images of high quality. These studies were classified in the literature based on different criteria, such as their need for a 3D model representation [105], their need for geometric information [106], and the number of the input images that are used in addition to the availability of geometric information [17]. In the following section we will illustrate briefly the underlying categories of each study.

3.5.1 Classification of the Reconstruction Methods

The first study[105] is classified into two main categories: 3D model-based methods; and 2D model-based methods.

- *3D model-based methods:* the virtual view is established through a complete representation of the 3D model. These methods only give a good result when a scene is simple; they don't work well in a complex scene. Moreover, the execution time is dependent on the scene complexity. A significant effort is required to produce an acceptable result; therefore, these methods are considered costly and time-consuming.
- *2D model-based methods:* these methods use a 2D model of the image, and do not need any 3D representation of the scene. In contrast to 3D model-based methods, the time required to generate new views using interpolation is independent of the scene complexity. These methods reconstruct the intermediate images based on the estimated disparity between the available known images.

Since the other two studies [17, 106] are based on similar criteria with little difference, we will combine their underlying categories:

- Many input images with known geometry

  Light-field rendering [107] and lumigraph [108] are two examples of this category. Since these methods do not need any geometric information, many input images are required to compensate for this unknown information. The intermediate images are produced by interpolating the existing sample of the images. Methods of this underlying category will suffer if the number of input images is few. The occlusion areas between the images are not a problem because the distances between the cameras are very small; therefore, the structure of the scene is visible in some of the existing images. Using many input images requires a lot of effort in performing the process; additionally, a large amount of storage space is required for the huge amount of redundant data.

- Few input images with known geometry

  Since the geometric information about the images is given, such as a depth map or 3D representation of the scene, the number of input images is reduced as in

[109, 110]. Using the available geometric information about the scene, the new virtual view is reconstructed. Under this category, the structure of occlusion area is a problem since few input images are available, although the location of such area is known due to the availability of the geometric information.

- Few input images with unknown geometry

  The reconstruction under this category is considered the hardest one, since no geometry information is available and few input images are available. Fortunately, the geometric information can be computed from the available images, which is called the disparity map. All the methods that estimate the disparity map in the reconstruction process are categorized under this category.

3.5.2 Classification of Intermediate View Reconstruction Methods Based on the Projection Direction

We will categorize the reconstruction methods based on how the disparity that compensated into the intermediate image is calculated; these are backward-disparity compensation methods and forward disparity-compensation methods.

3.5.2.1 Methods that use backward-disparity compensation:

1. Backward projection intermediate view reconstruction algorithms

In backward-disparity compensation methods, the disparity values are defined on the sampling grid of the intermediate image (unknown images). Because the intermediate image will point to the irregular sampling grid of the known images, spatial interpolation is used. These backward-projection algorithms use two different ways to reconstruct the intermediate-view points from different known images; these are linear interpolation [73], and non-linear interpolation [75].

Using linear interpolation, the intensities of the intermediate image will be calculated by weighted averaging of the intensities of the left and right images, as in equation (3.5), while the methods that use non-linear interpolation reconstruct the intermediate image either from the left or right image.

Mancini and Konrad [73] proposed a reconstruction algorithm based on the quadtree disparity estimation which uses the linear-interpolation to reconstruct the

intermediate view. However, the main purpose of this paper is to enhance the block-based disparity method by changing the block size at object boundary to the smaller one; after that the intermediate image is found. Since this method uses linear interpolation, a blurred image will be produced for the reasons mentioned in section 3.3.1. Thus, the blurriness and occlusion problems are not addressed in this paper. Therefore, a reconstruction algorithm based on a winner-take-all strategy is proposed [75] to address the blurriness problem of the linear interpolation. This algorithm is based on a fixed-block size disparity estimation algorithm, where the intermediate image is reconstructed by compensating appropriate blocks either from the left or right image. Consequently, a patchy effect will be produced. In addition, the disparity values, the intensities of the intermediate image, and the decision field are calculated jointly by a Bayesian formulation and solved by Maximum A posteriori Probability (MAP) estimation, which makes this method computationally expensive. In addition, this algorithm is not aware of the occlusion area.

Later on, the Philips research group developed a motion compensation interpolation to create intermediate views in real time [74]. This algorithm is a backward-disparity compensation that uses linear interpolation by averaging the intensities of both the left and right images. A disparity map is estimated using a 3D Recursive Search (3DRS) algorithm, which is an efficient motion estimation algorithm developed specially for the motion compensation interpolation. Since the 3DRS algorithm is a block-based technique that assigns one motion vector for all pixels in the block, a block erosion approach is used to find a specific motion vector for each pixel in the block. Thus, the 3DRS algorithm becomes a pixel-based approach rather than a block-based approach. As the 3DRS algorithm uses a temporal candidate vector when computing the motion vector, which comes from a previous frame, a temporal consistency is ensured. Hence, any flickering artifices will be reduced to generate a smoother disparity map. Also, the occlusion problem is not addressed.

Recently, an improvement to the backward-projection algorithm was proposed [17] by preserving the discontinuity at the edges to overcome the blurring effect of the linear interpolation. In addition, the occluded area is handled since no backward-projection algorithm in the literature take into consideration the occlusion problem.

Ince *et al.* [17] implement two algorithms, one to reduce the blurriness effect and the other one to handle the occlusion problem. In the first one, they use two stereo images to reconstruct the intermediate image by the standard backward-projection algorithm, and they use this reconstructed image as an input to the edge-preserved disparity regularization (*i.e.* anisotropic regularization) to reconstruct a more accurate intermediate image without blurriness at the edges. This is because they detected the edges in the intermediate image and found the edge maps to be very similar to the original intermediate image.

The second algorithm uses four input images instead of two images to handle the occlusion problem in addition to applying the first algorithm to preserve the discontinuity at the edges. Since they use four input images, the structure of the occlusion area is visible in at least two different images. They use a variational formulation to calculate the disparity from multiple images. The occluded area is detected using the improved geometric approach [88]. Moreover, it can be used to determine from where each image point in the intermediate image can be compensated, which is formulated in a variational formulation. The necessity of using many input images and the cost of the computation are considered major drawbacks of this algorithm. Therefore, a robust and efficient backward-projection algorithm is required.

2. Backward projection Depth Image-Based Rendering (DIBR) algorithms

Another way to find an intermediate image using backward projection is Depth Image-Based Rendering (DIBR), which warps the coordinates of the reference images onto the virtual image using a depth map. Most of these algorithms generate two virtual images from the available reference images and their depth maps to blend them into one final image, [72, 111], while others use a single reference image with its depth map to generate a virtual image [112]. With the former algorithms, the disocclusion areas are handled more accurately than when using the latter algorithms; this is because there is more chance of recovering the disoccluded area from both reference images. The well-known framework for the DIBR algorithms that use two reference images is depicted in Figure 3.5.

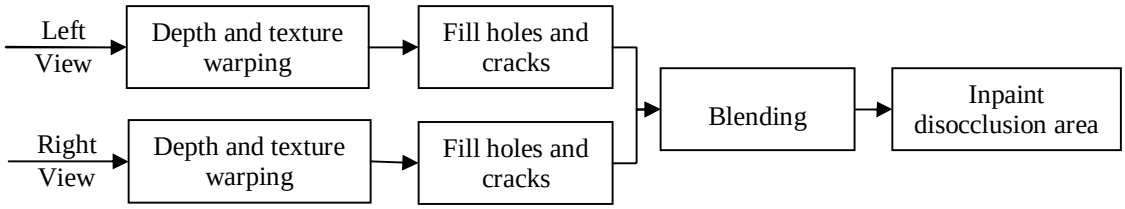


Figure 3.5: The general framework for DIBR algorithms

Most of the DIBR algorithms consist of four main steps: warping, filling holes, blending, and disocclusion inpainting. DIBR algorithms differ in their ordering of the interpolation steps, as some algorithms warp the depth map and the texture simultaneously as one step, and other algorithms warp the depth map first to employ it in warping the texture of the reference image to the virtual one. Unfortunately, generating two virtual images that blend finally into one image is considered a redundant operation that adds extra computation to the whole process of rendering algorithms. Specifically, if we need to calculate ten virtual images in a stereoscopic system for a multiview display, in the rendering process it will be performed twenty times: twice for each image.

The MPEG group developed a DIBR algorithm that finds a virtual intermediate image between two reference images using their depth maps [111]. Firstly, they calculate homography matrices that specify the image relationship coordinate between the reference and virtual view using the available geometry information in the projection matrix (position, rotation …). Then, the depth map of the virtual view is calculated depending on the calculated homography matrix; after that, the texture of the virtual view is created from the available information of the reference view and the inverse of the homography matrix. Ghost contours are removed by dilating the uncovered unknown areas 1-pixel width to be recovered from the other virtual image generated by the other reference image. These two generated virtual images are blended into one virtual image by using bilinear interpolation that contains holes and cracks. These holes are filled from neighbouring pixels. The general framework of this method is depicted in Figure 3.6.

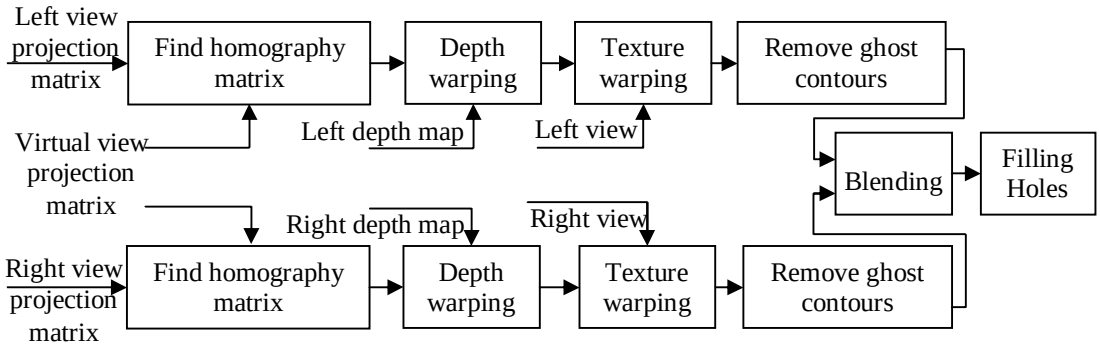


Figure 3.6: The proposed DIBR framework by MPEG group[111]

Unfortunately, calculating the homography matrix adds extra complexity to the DIBR algorithm; in addition, these parameters are not always available. Also, the general holes and cracks on the virtual depth map are not filled before the virtual view generation as it may be filled incorrect information. Moreover, the post-processing hole-filling procedure does not recover the information accurately since it does not take into consideration the object boundaries.

Mori *et al*. proposed a DIBR algorithm that recovers the drawbacks of the MPEG DIBR algorithm [113] as they do not use any geometric information in calculating the virtual depth map. After warping the depth maps of the nearest reference images, the holes and cracks are filled by using a median filter. After that, a bilateral smoothing filter is applied to reduce depth map noise while preserving the edges. Then, a virtual image is generated from each reference image by backward-projecting the intensities of the reference images where these virtual images blend into one virtual image. The resulting ghost contours that appear in the blended image are removed by boundary-matting, which requires the extension of the occlusion areas from the background direction to be copied from one reference image. Disocclusion areas are filled by applying an inpainting algorithm proposed by Telea [114]. The general framework for this DIBR algorithm is depicted in Figure 3.7.

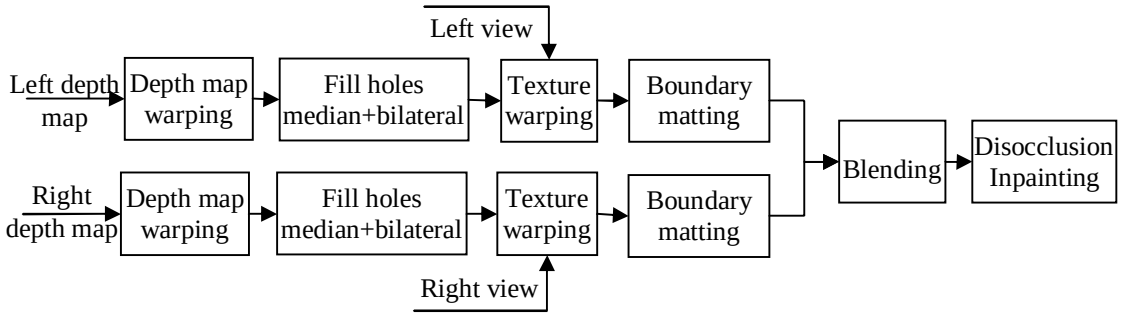


Figure 3.7: The proposed DIBR framework by Mori *et al.* [113].

Unfortunately, the inpainting algorithm adds an extra computation complexity since image-inpainting techniques are computationally expensive. In addition, this inpainting technique is not sufficiently specialized to fill the disoccluded areas. So, the filled areas are not accurate.

Later on, another DIBR algorithm was developed by Jeong *et al.* [112] which generates a virtual view from one reference image and its estimated depth map. The generation process concentrates mainly on depth map-filtering and determining how to recover the disocclusion areas as shown in Figure 3.8. A discontinuity-preserving smoothing filter is used to reduce the depth map noise by smoothing the object area itself while preserving object boundaries. The disocclusion area is estimated by using geometrical information on the views, where it is filled from the neighbouring frames (temporal information) by matching image blocks. In this algorithm, the ghost contours are not handled, which has a noisy effect on the virtual image.

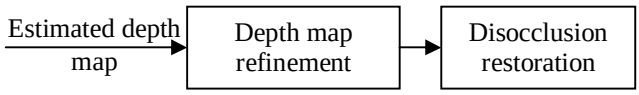


Figure 3.8: The main steps of Jeong *et al*. DIBR algorithm [112]

At the same time, Do *et al*. changed the order of the warping steps to reduce depth map errors [115]; they warped the texture and the depth map simultaneously in the same step instead of using the warped depth map of the virtual view to warp the texture of the reference image. The generated holes and cracks on the virtual depth map are filled using a median filter, and the coordinates of the ghost contours are

removed from the virtual depth map after 3D-warping the discontinuities' coordinates. Then, the two generated images are blended into one image by weighted averaging of the two nearest reference images. The general framework of this method is depicted in Figure 3.9.

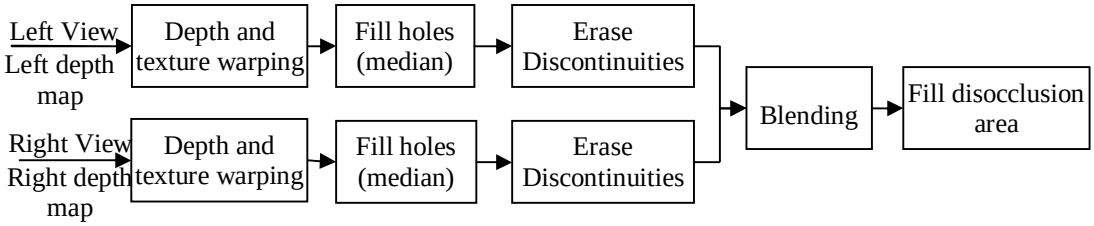


Figure 3.9: The proposed DIBR framework by Do *et al.* [115]

As a post-process, the disoccluded areas are filled from the nearest background intensities; this results in blurring in the inpainted area as well as inaccuracy in the filled information. Consequently, Do *et al.* improved the inpainting technique described above by considering the depth information while filling the disoccluded areas [72]. For each pixel, a weighted averaging for the nearest eight background pixels is calculated. Unfortunately, this technique is not sufficiently robust to fill large occlusion areas; also, it does not preserve the discontinuities of the object's edges during the filling procedure. In this DIBR method, the pixels that cause ghost contours are labelled in the reference image to be omitted from the warping step. After that, the depth map and the texture values are warped into a virtual view in the same step. The generated holes and cracks in the depth map are filled by a median filter and, at the same time, the indexes of the filled holes are copied to inversely warp the holes from the virtual image to the reference image. A blending step is required to generate one virtual image with a disocclusion area which is inpainted by the improved inpainting technique described above. The general framework for this DIBR algorithm is depicted in Figure 3.10.

Very recently, the steps of the DIBR algorithms have been analyzed in detail to determine the contribution of each step in the improvement process to the quality of the final image in [116]. The results show that the warping step contributes greatly to the quality of the virtual image. Consequently, further analysis was carried out in [117] to determine exactly where the errors in this step are produced and two reasons

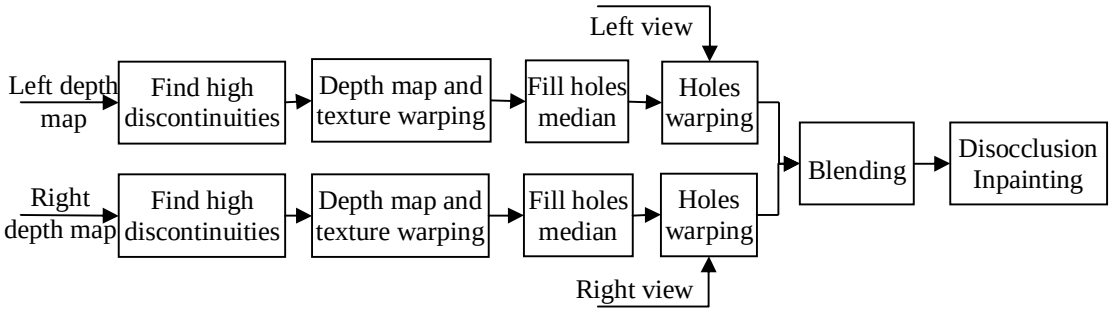


Figure 3.10: The general framework of the proposed improvement techniques on the DIBR algorithm by Do *et al*. [72]

for these errors were found. The first one is generated from rounding errors in the sampling grid of the virtual image which are suppose to be integers; the second source is insufficient depth quantization. In [117], the rounding is solved by proposing a supersampling warping approach, and the authors apply a supersampling with a factor of two to warp the image points. Although the supersampling reduces the holes generated in the virtual image and the number of rounding errors, it is considered computationally expensive as each pixel in the reference image will be warped four times. Moreover, the virtual image needs to be downsampled to its original size, and some errors will still remain in the image.

Meanwhile, an asymmetric DIBR algorithm has been proposed; this reduces the image artifices depending on the assumption that was proposed in [118]. This assumption assumes that, when one of the stereo pair is the original or close to the quality of the original image and the other one is a slightly corrupted image, the perceived quality of the stereoscopic image will not be affected by the corruption that exists in one of the images. They assume in their algorithm that the left image is always the original image and the right image is the interpolated one. The texture of the virtual image is calculated by backward projection into the available reference images. Subsequently, image artifices are detected by applying a Laplacian filter to generate a confidence map; then the anisotropic diffusion filter based on the Perona-Malik equation [119] is applied to smooth out these artifices from the virtual image. This method is effective for small artifices but is not reliable for large areas of artifices. Moreover, this algorithm allows us to generate just one image between every two reference images, which is not a practical technique in reality.

3.5.2.2 Methods that use forward-disparity compensation

In contrast to the backward-disparity compensation, the structure of the intermediate image using the forward-disparity compensation approach is defined on the sampling grid of one or all of the available images. The problem occurs when the available images are pointing to the irregular samples of the intermediate image. Simple and complicated solutions are suggested in the literature to solve the problem.

McVeigh *et al.* [76] proposed a block-based reconstruction algorithm that estimates the disparity values using a block-matching approach of fixed-size blocks. After that, the intermediate image is reconstructed using forward-disparity compensation. The irregular-point problem is avoided by using a full-pixel precision. The occlusion area is detected and handled using a depth constancy assumption that assumes the depth will be the same within a small area of neighbours, although this assumption is not always valid.

Scharstein [77] used the forward-disparity compensation to reconstruct two intermediate images, one pivoted from the left image and the other one from the right. The final intermediate image is a result of blending these two intermediate images. The problem of the irregular sampling intensities is solved by rounding them to the nearest integer, which of course distorts the quality of the intermediate image. This method depends on the rectification process of the stereo images which helps the disparity estimation and intermediate view reconstruction processes. The disparity map is estimated using an extension to the intensity gradient method [120], which explicitly handles the occlusion problem. The occluded area is detected based on the mismatch between the left-to-right and right-to-left disparity map. Scharstein refers to the handling of the occlusion area and the newly-exposed area problems as the visibility and filling holes problems, respectively. The visibility problems occur when two or more image points are mapped to the same location in the intermediate image; this is resolved by remapping the image points in their correct order to their new location. The holes problem occurs when some points in the reconstructed image are not assigned any intensity. Some of these holes are filled automatically when blending the two intermediate images, and the other non-filled holes are handled using the texture synthesis, which is a problematic solution in detailed areas.

Since the intermediate image is always reconstructed in between two images, Redert *et al*. [121] proposed an algorithm to find the intermediate image at non-intermediate positions. This is performed by generating an intermediate image once at the central position between the two images. After that, any new virtual image can be extrapolated using this central interpolated intermediate image and a single right-to-centre or centre-to-left disparity field. The block-based dynamic programming method is used to estimate the disparity map. The occlusion and newly-exposed areas are called overdefined and undefined points, respectively, in this paper. The overdefined points occur when two or more image points in the centre are assigned to the same location in the new reconstructed image, whereas the image points that do not have any intensity values are called undefined points. The former are solved by choosing the closest point to the new virtual image, while the latter are filled by linear interpolation of the neighbouring image points.

Generating an arbitrary view using more than two images is proposed in the literature [91]. Park and Inoue used a five-camera system, where one of them is located in the centre location and the others are located to the left, right, above, and below the centre one. A hierarchical disparity-matching algorithm [122] is used to estimate the disparity values. After that, the depth map is estimated for the centre image and then forward-mapped to the new virtual image. Overdefined and undefined points are found in the virtual image due to the forward-mapping. The overdefined point problem is solved by choosing the smallest depth value (i.e. depth constancy assumption), while the undefined points, which the authors call 'the uncovered area', are filled by the Observable Viewpoints (OVP) concept. The OVP fills the undefined points based on the direction of the mapping where they are located; in this way, they are filled by the maximum depth values. If all of these assumptions fail, the texture synthesis algorithm is used to fill these areas.

Recently, a spline-based reconstruction algorithm has been proposed to solve the problem of irregular-sample space through the reconstruction process [71]. A spline-based algorithm assigns a unique intensity to each image point in the virtual image; consequently, the overdefined and undefined points will not arise again in the virtual image. This method is based on the minimization of the energy function which

consists of a data-matching term for a B-spline model of the irregular image samples and a B-spline smoothness term. Although this minimization gives up a continuous function that recovers the regular sample space for the intensities, it is computationally expensive, which is considered a drawback for this method. This algorithm jointly performs the optical flow disparity estimation, the occlusion detection, and the extrapolation of the occlusion area in one formulation [93]. The extrapolation is performed by the anisotropic diffusion which is guided by image gradient. The interaction between the disparity estimation and occlusion detection is advantageous in this approach. Unfortunately, the computation cost is very expensive and it is more sensitive to disparity estimation errors than the backward-projection algorithm.

Finally, we will summarize all of the above algorithms in Table 3.1 and the essential steps for generating an intermediate view with high quality in Figure 3.11.

Table 3.1: Summary of the prior works on the intermediate view reconstruction

Method Name	Disparity Method Name	Reconstruction Method Name	Occlusion Detection / Handling/	Characteristics
Quadtree block-matching Mancini & Konrad (1998) [73]	– Block-based matching (variable-size block)	– Backward disparity compensation (linear interpolation)	– Not addressed (N/A)	– Adaptive window – Large block→non-boundary – Small block → boundary – Blurriness
Winner-take-all approach Mansouri & Konrad (2000) [75]	– Block-based matching (Fixed-size block)	– Backward disparity compensation (non-linear interpolation)	– Not addressed (N/A)	– Decrease the blurriness problem of pivoting – Generate patchiness effect
Philips research group (2006) [74]	– 3D Recursive Search (3DRS) method (Motion estimation)	– Backward disparity compensation (Motion Compensation interpolation)	– Not addressed (N/A)	– Since 3DRS algorithm use temporal candidate, the temporal consistency is ensured→smoother disparity map
2-images isotropic backward	– Variational formulation	– Backward disparity compensation using	– Not handled	– Reduce the blurriness effect, by this the

projection Ince & Konrad (2008) [17]	using anisotropic regularization(coarse-intermediate image)	variational formulation		edges are preserved
4-images Occlusion- Aware Backward projection Ince & Konrad (2008) [17]	– Variational formulation using anisotropy regularization (coarse-intermediate image)	– Backward disparity compensation via variational formulation	– Using 4 images provides sufficient information about the occlusion area which is detected using an improved geometric approach and handled using anisotropic diffusion (via variation formulation)	– Reduce the blurriness effect, by this the edges are preserved – Because every step is performed via a variational formulation, this improvement to the backward projection is considered very expensive.
MPEG Group Gotfryd et al. DIBR method (2008) [111]	– Estimated depth map	– Backward disparity compensation (using homography matrix) – Blending the generated two virtual images using bilinear interpolation	– The generated holes and cracks in the final image is filled from the neighbouring pixels	– The rendering is done twice; one for each reference image. – Calculating homography matrix adding an extra complexity – Inaccurate filling procedure.
Mori et al. DIBR method (2008) [113]	– Estimated depth map	– Backward disparity compensation – Blending the generated two virtual images using alpha blending.	– The disocclusion area is inpainted by Telea inpainting algorithm [114]	– The holes and cracks in the virtual depth map are filled by median filter – The bilateral filter is applied to reduce the noise in the depth map. – The ghost contours are removed by boundary matting. – Telea inpainting technique is computationally expensive and it does not take into consideration depth information.
Jeong et al. DIBR method (2009) [112]	– Estimated depth map	– Backward disparity compensation	– The disocclusion area is estimated using the	– Discontinuity preserving smoothing filter is used to

			geometrical information of the views. – It is filled from the neighbouring frame (temporal information).	reduce depth map noise by – →smoothing objects →preserve boundaries – The ghost contours are not handled –
Quality improving techniques in DIBR Do et al. (2009) [115]	– Estimated depth map	– Backward disparity compensation – They warp the depth map and the texture simultaneously – Blending the generated two virtual images using weighted averaging.	– The disoccluded area is filled as a post-process from the nearest background intensities	– The holes and cracks in the virtual depth map are filled by median filter – The ghost contours are removed – Blurriness in the inpainted area in addition to the inaccuracy.
Quality improving techniques in DIBR Do et al. (2010) [72]	– Estimated depth map	– Backward disparity compensation – Blending the generated two virtual images using alpha blending.	– The disocclusion inpainting technique takes into consideration depth information. – They are filled by weighted averaging the nearest eight background pixels.	– The holes and cracks in the virtual depth map are filled by median filter – The ghost contours are omitted from warping. – Noisy cut-off effect at object boundaries from omitting the ghost contour warping. – The inpainting technique is not robust for large occluded areas, also it does not take into consideration the continuation of edges.
Supersampling warping method Do et al. (2011) [117]	– Estimated depth map	– Backward disparity compensation – They apply the same method of Do et al. (2010) [72] – They apply supersampling of a factor two to warp	– Same of the inpainting method of Do et al. (2010) [72]	– Analysing the warping step is performed. – Most of the errors in the warping step are generated from rounding the sampling grid.

		the image points.		– Although supersampling reduces rounding errors it is computationally expensive. – Each pixel with supersampling is warped four times than without supersampling. – Extra efforts are performed to downsample the virtual image to the original size.
Asymmetric DIBR method Devernay et al (2011) [118]	– Estimated depth map	– Backward disparity compensation – They assume that if one of the stereo images is original image and the other one is slightly corrupted, the perceived quality will not be affected.	– It is considered an image artifice and it is detected by Laplacian filter then the anisotropic diffusion filter is applied to smooth out these artifices from the virtual image.	– They assume that the left image is always the original image and the right image is the interpolated one. – →They can interpolate few images between two reference images. – Image artifices are detected by Laplacian filter, and smoothed out by anisotropic diffusion →not effective for large areas like occlusion area –
Mc Veigh et al. (1996) Method [76]	– Block-based matching (fixed-size block)	– Forward disparity compensation	– Handling the occlusion using depth constancy assumption	– Use full-pixel to avoid the irregularly spaced intensities
Scharstein (1996) Method [77]	– Intensity gradient method [120]	– Forward disparity compensation. Two intermediate images are forward mapped from left and right images, then they are combined by using weighted averaging	– Rectification and disparity estimation are used to handle occlusions – Overdefined points solved by remapping the points in their correct order	– Simple – Rounding the pixel position to the nearest integer (result which degrade the quality of images) – Newly exposed area (holes) is filled by using texture syntheses algorithm

			– Holes filled by texture synthesis	which causes problems in detailed areas.
Redent *et al.* (1997) Method [121]	– Dynamic programming	– Forward disparity compensation. Compute the intermediate view at the centre position between two cameras, then use this centre view and single disparity map to map any virtual view	– Occlusion is handled by choosing the closest point to the virtual image. – Newly exposed area are filled by linear interpolation of neighbouring intensities	– Can reconstruct views at non-intermediate positions – It is successful in the simple scenes structure –
Multi-camera methods Park & Inous (1997) [91]	– Hierarchical-disparity estimation [122]	– Forward disparity compensation. Arbitrary view generation using five cameras (centre, above, below, right, left) – forward mapping the depth map of central camera to that of virtual camera	– Using many cameras may better define the occluded areas – They used the depth consistency and other assumptions to fill the uncovered area – If all of them fail they use the texture synthesis	– Overdefined and undefined points are found due to the forward mapping – For undefined points: they use the observable viewpoints (OVP) concept for to fill the holes based on the direction of mapping – For over defined points choose the smallest value among multiple depth value – In case of all assumption failing, they use the texture synthesis which fails in detailed areas.
Spline-based intermediate view-reconstruction Ince & konrad (2007) [71]	– Optical flow [93]	– Forward disparity compensation. – B-Spline reconstruction method	– Detecting & handling jointly with the disparity estimation in one formulation (detected using the gradient of the underling images and the occluded area are extrapolated using anisotropic diffusion)	– Solve the irregular-sample space that occurs in the reconstruction process. – The overdefined and undefined points will not arise again in the virtual image. – Spline-based reconstruction algorithm needs a lot of computation

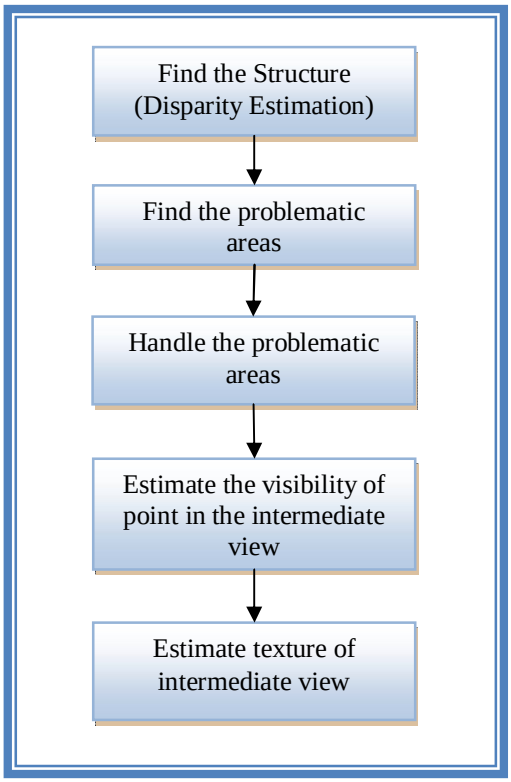


Figure 3.11: The essential steps for generating an intermediate view of high quality

3.6 Summary

In this chapter, we explored the challenges of intermediate view reconstruction in detail. Then, previous solutions to these challenges were presented. Prior classifications of the intermediate view reconstruction algorithms were presented; these differ from our classification. We classified the prior work on intermediate view reconstructions based on the projection direction, which might be backward projection or forward projection. Our work is classified under the backward projection category. Depth Image-Based Rendering (DIBR) is one of the approaches that are used for the reconstruction process; instead of using disparity to calculate the virtual image, this uses the depth map, which needs more geometric information on it.

To conclude, the existing work by other researchers on generating the intermediate views are still generating image artifices due to the difficulty of occlusion area restoration, inaccuracies of disparity maps, and projection errors. Therefore, there is still room for further improvement by finding a new reconstruction algorithm that

will find intermediate images of high quality by handling the occlusion area and preserving the discontinuity at the edges. In this chapter we explored intermediate view challenges; in the next chapters we will propose novel algorithms that handle most of these challenges, and we will employ them in highly robust intermediate view reconstruction.

CHAPTER

4

Newly-Exposed/Occlusion Detection (NEOD) Algorithm

Placing objects in different positions in 3D scenes leads to the visibility of some regions in one view and their disappearance in the other; such regions are known as occluded areas. Computer simulation and experimental results confirm the importance of occluded areas in depth perception, disparity estimation and intermediate view reconstruction, so finding an accurate occlusion detection algorithm is highly desirable. Many algorithms have been proposed to detect the occlusion areas but not to the level required for robust detection; furthermore, they are computationally too expensive. Therefore, in this chapter, we propose a Newly-Exposed/Occlusion Detection (NEOD) Algorithm that detects the newly-exposed and occlusion areas accurately without noise.

This algorithm is simple and efficient and gives accurate results; it is based on the physical displacement of the objects between the stereo images without taking into consideration image intensities. A comparison with three well-known region-based occlusion detection algorithms is performed under accurate and inaccurate disparity quantitatively and qualitatively.

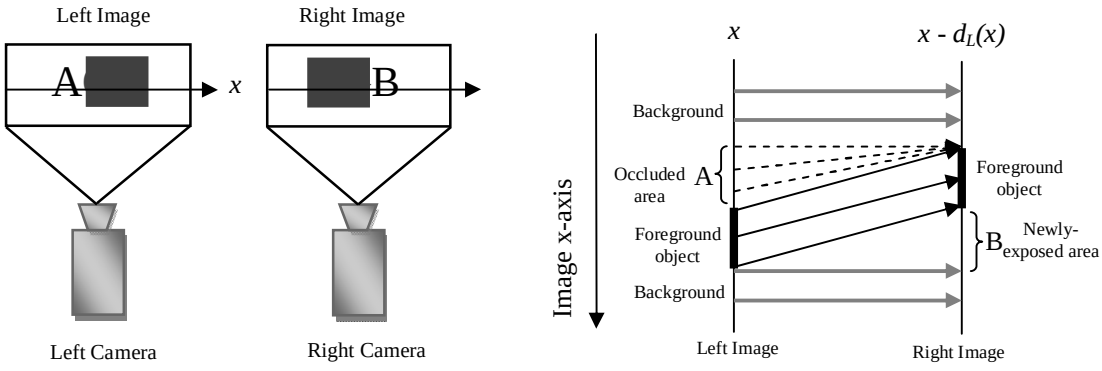
4.1 Introduction

In the physical world, when objects are placed at different distances, some regions will be visible in one eye but occluded from the other [123]; such a region is called a *monocular zone*. Many studies have been conducted to obtain evidence that a monocular zone provides a good source of depth [123], and others have proved that such areas improve the depth perception itself [124]. Therefore, it is very important to consider these zones in stereoscopic displays. With stereo images

the monocular zones are called *occlusion* and *newly exposed areas* that arise from the capturing of images from different positions.

The occluded area is defined as the visibility of such an area in one image and its disappearance in the other, depending on the direction or ordering of the camera. On the other hand, if the disappearing area becomes the visible area, this area will be called the newly-exposed or uncovered area. Figure 4.1 shows the original stereo pair of left image I_L and right image I_R. As we can see, the area "A" in I_L is disappearing in I_R, known as the occluded area, and "B" in I_L is appearing in I_R, known as the newly-exposed area. These definitions are also illustrated in Figure 4.2 (c); this synthetic scene is found to highlight the occluded and newly-exposed areas, which are marked in black and grey, respectively, if they are ordered from left-to-right as depicted in the Figure 4.2.

Since the stereo images result from projecting the same 3D scene in the physical world from different positions, the 2D image points in the stereo images are corresponding but in different places. The recovery process for these corresponding points in the stereo image pair is called disparity estimation. Generating additional



(a) (b)

Figure 4.1: Monocular regions, from left-to-right: "A" is occluded area and "B" is newly-exposed area

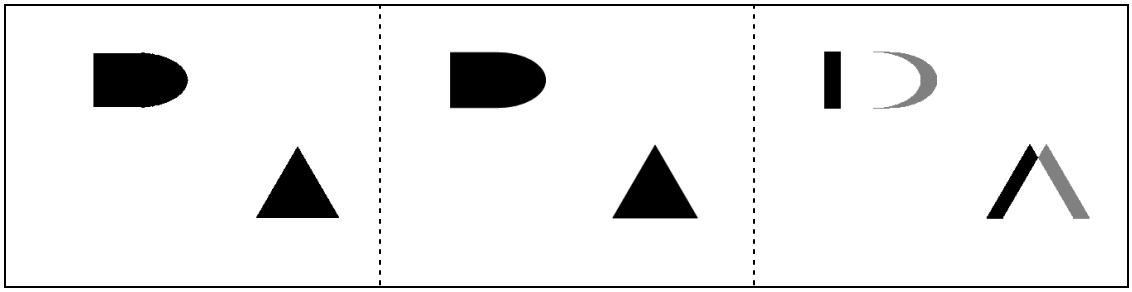


Figure 4.2: Monocular region, from left-to-right: black area is occluded area and grey area is newly-exposed area

views between the stereo image pair by positioning virtual cameras to create virtual views at that position is called intermediate view reconstruction. Occluded areas have an impact on these two problems, disparity estimation and intermediate view reconstruction. Because the disparity estimation depends on the intensity matching, and the occluded areas are defined as visibility of the image points in one image but not in the other one, the correct disparity value will not be calculated in these areas. On the other hand, the quality of the intermediate image depends on the accuracy of disparity and it is difficult to calculate an accurate disparity map due to occlusion areas; thus the occlusion area will affect the reconstruction process.

In the next section, we demonstrate some of the existing works on the occlusion detection, and give more details of a number of selected algorithms for comparison purposes. Then, in section 3, the proposed algorithm is presented. The experimental results are analyzed in section 4, with the conclusion set out in section 5.

4.2 Background

Since the occlusion area is considered problematic in many applications, several studies have been carried out to detect this area. Detecting an occlusion area using more than two images has been proposed in the literature [78], but it needs an extra computational effort and a larger amount of data. Some approaches and constraints have been proposed in the literature to detect the occlusion area itself or the occlusion border, but most of them fail to give a reliable result or/and are computationally expensive. An empirical comparison between five well-known

occlusion detection approaches has been conducted under two disparity matching algorithms [90]. Bimodality (BMD) [125] and Match Goodness Jumps (MGJ) [126] detect the occlusion borders, i.e. where they begin and end, and the other three, Left-Right Checking algorithm LRC [79-81], Ordering constraint ORD [82, 83], and Occlusion constraint OCC [83, 84], find the whole occlusion region. LRC algorithm uses the intensity matching between stereo pairs as an indicator of the occlusion area as they are referring to the same scene, while ORD constraint assumes that any disordering of the corresponding image points of the stereo pair is considered an occlusion area. Furthermore, the occlusion area is detected using the OCC constraint based on the principle that such an area leaves unmatched points in the other image; consequently it will make a jump near to the boundary. This comparison finds that each method works in a different way depending on the image textures (simple or complex) and the accuracy of the disparity matching algorithm.

Another constraint is the uniqueness constraint which checks whether each image point in the first image has at most one corresponding match in the second image [85]. Matching between left-to-right and right-to-left disparity maps can be used to check one-to-one matching points. The disparity estimation methods apply a smoothness constraint to the object's surface in the image except at the object boundaries [85]. Thus, the object surface has a small gradient while the object boundary has a high gradient [86]. Since the occlusion areas are located near to the object boundaries, they will have a high gradient. Therefore, the smoothness and gradient constraints can be used as indicators of occlusion areas.

The visibility constraint [87] is another technique used to detect the occlusion area; the idea of this constraint is obtained from the occlusion concept which ensures that the visible pixels have at least one match on the other image while the occluded pixels have no matches. Consequently, this constraint finds the newly-exposed area that has no matches in the other image. Thus, the visibility constraint needs to find the occlusion area in one image from the disparity of the other image to ensure that they are consistent. Since the visibility constraint allows many-to-one matching as shown in Figure 4.3, this gives it more flexibility, although this is sometimes considered the weakness of this constraint. Related to this concept, a geometric-

based approach [88] has been developed to detect the newly-exposed area by measuring the spatial density of projections in the target image after forward projection of the disparity of the reference image into the target image. The area that is highly referenced is considered an occlusion area, as in Figure 4.3, whereas an empty area that does not have a relationship with the reference image is a newly-exposed area. This approach detects the empty area from projections, which is a much easier process.

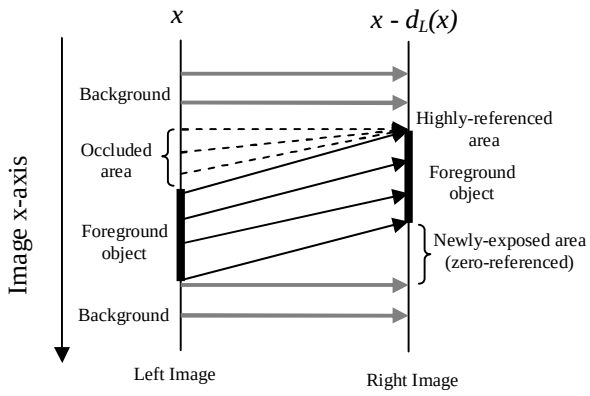


Figure 4.3: Forward-projection from reference image (left image) to the target image (right image) that forms occluded and newly exposed area: solid and black lines are the foreground object, solid and grey lines are the static background, dotted lines are occluded area, and the empty area is newly-exposed area.

Recently, Phillips [89] have exploited the fact that the occlusion areas are characterised as predominantly located near to the image boundaries; thus, the quality of the matching metric at these boundaries will decrease suddenly. Since they use the absolute difference of the corresponding image points of the stereo pair as a matching metric which is considered a noisy map, they apply a median filter as a post-processing step to reduce this noise. Then they apply an adaptive boundary-preserving filter to highlight the occlusion area at the discontinuity of the smoothed matching metric.

Most of the occlusion detection methods that depend on calculating matching errors need to select a certain threshold for comparison purposes to decide whether it is an occlusion area or not. Unfortunately, this selection process is very sensitive and it is difficult to make choices since each image might need a different threshold

depending on the amount of texture details. Moreover, with the presence of varying texture complexity per image, applying one threshold per image will not allow all the occlusion areas to be detected; alternatively, they will be falsely labelled as occlusion areas. Since state-of-the-art methods that use thresholds to detect occlusion areas are very sensitive to the image nature (i.e. simple or complex), and methods that use intensity matching do not give reliable results in the presence of image noise, a more reliable occlusion detection method is required.

In this chapter, we propose a novel occlusion detection algorithm which does not depend on the intensity matching and can detect the occlusion area without using any threshold. This novel algorithm is based on a physical displacement of the objects between the stereo pair, so the detection process is performed upon the recovery process for the displaced object position, i.e. by checking whether the place of the displaced object is covered or still uncovered. In addition, we compare the proposed algorithm to the three well-known region-based occlusion detection algorithms, LRC, ORD and OCC, which are empirically compared in the literature [90]; this is because they detect the whole occlusion region as the proposed one. As the quality of the disparity matching algorithm depends on the occlusion results, we test all algorithms on a ground truth disparity to show the accuracy of each algorithm because, if the result is not accurate under ground truth, what will it be when using inaccurate disparity? Moreover, the proposed algorithm is tested using disparity calculated by the cooperative optimization for region-based matching algorithm [127] to show what it will be under non-ground truth disparity. We will provide more details in the following subsections on three well-known occlusion algorithms in the literature (LRC, ORD, and OCC) for comparison purposes.

4.2.1 Left-Right Checking Failures Algorithm (LRC)

This approach uses intensity matching error as an indicator for the occlusion area. If a certain pixel intensity in the first image does not match the corresponding pixel in the second one, it is because it does not exist in the second image; consequently it is labelled as occluded. This intensity matching error is measured by the following:

$$E_{LR}(x) = I_L(x) - I_R(x - d_L(x)), \tag{4.1}$$

$$E_{RL}(x) = I_R(x) - I_L(x + d_R(x)), \tag{4.2}$$

where I_L and I_R are image intensities in left and right images, respectively, d_R is the disparity from right to left, while d_L is the disparity from left to right, E_{LR} is the resulting matching error from left to right image, and E_{RL} is the resulting matching error from right to left image.

The resulting error values for each pixel's intensity are compared with a certain threshold, which has been chosen empirically, to determine whether a pixel is labelled as an occluded or visible pixel. Unfortunately, the intensity matching error may result if there is a noise in the image or the illumination of the specular surface is changed; in this case the photometric approach will become inaccurate.

4.2.2 Ordering Constraint (ORD)

The ORD constraint assumes that the order of the pixels in the same row is not changed between stereo image pairs, unless an occlusion area is found [82]. The Figure 4.4 below illustrates how the order of objects is different when captured from the left and right cameras. The foreground object C blocks some part of the background visibility. In particular, the foreground object C blocks the visibility of D area in the left camera and B in the right camera.

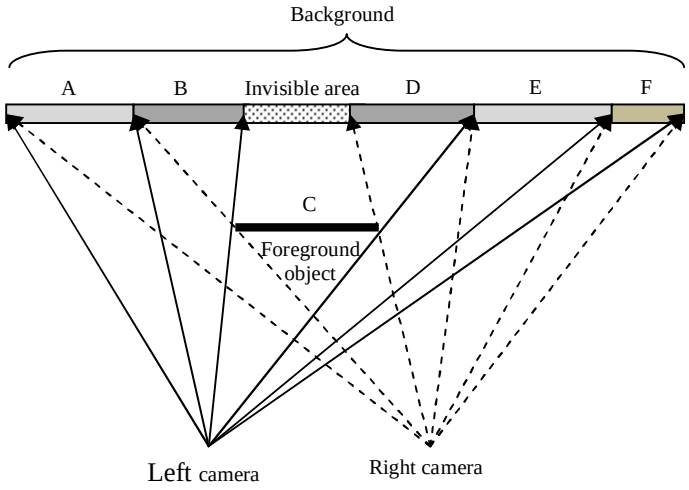


Figure 4.4: Ordering the foreground object and the background in two layers to demonstrate the visible and the invisible areas in both left and right cameras.

Table 4.1 shows two examples derived from Figure 4.4 to demonstrate how the ordering constraint works. In the first row, the order of pixels is different between left and right images; B is occluded in the right and D is a newly-exposed area in the left, while the order of pixels EF in the left image is preserved in the right image. This difference in order refers to the occluded area. The detection process of order is achieved by computing the intensity matching errors for each group of pixels in the same line. As in row1 in the Table 4.1, the order is detected for each of the three pixels by calculating three matching errors for each pixel as in equations (4.3), (4.4) and (4.5), where *x, x+1, x+2* are the indices of A, B, and C respectively. After that, the resulting errors are compared with a certain threshold, which has been chosen empirically, to determine whether it is occluded or not.

$$E_{LR}^{1}(x) = I_L(x) - I_R(x - d_L(x)), \tag{4.3}$$

$$E_{LR}^{2}(x+1) = I_L(x+1) - I_R((x+1) - d_L(x+1)), \tag{4.4}$$

$$E_{LR}^{3}(x+2) = I_L(x+2) - I_R((x+2) - d_L(x+2)), \tag{4.5}$$

Table 4.1: Demonstrates the order of objects in the left and right images projected from the left and right cameras as in Figure 4.4 and illustrates their ordering and occlusion.

Left Image	Right image	Ordering	Occlusion
ABC	ACD	Not ordered	B is occluded in the right, D is occluded in the left
EF	EF	Ordered	Not occluded

Unfortunately, this constraint fails in the presence of thin objects, where the order is not preserved even it is not an occluded area; this is illustrated in Figure 4.5. Figure 4.5 (a) shows that the left image has two dots (refer to the dot A) located to the left of the two thin bars (refer to the bar B) where the order is (ABAB). This order is

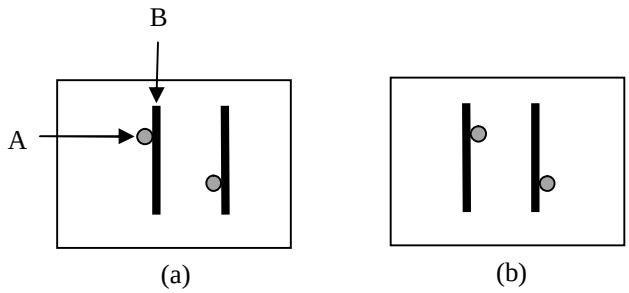


Figure 4.5: (a) Left image, (b) Right image. Erroneous ordering constraint in presence of thin foreground object (thin bar B): the order of object B and the dot A is not preserved in left and right images.

changed in the right image to become (BABA), as the dots appear to the right of the thin bars, as shown in Figure 4.5 (b).

4.2.3 Occlusion Constraint (OCC)

Since the occluded points are visible in one image and not in the other, one of the stereo pair will skip over a certain point during the matching process leaving unmatched points in the other image, which is labelled the occlusion area. As this jump will be near to the boundaries, the occlusion boundary is determined by equation (4.6) or (4.7). This constraint is unlike the LRC and ORD since it does not depend on the intensity matching, where the occlusion boundaries are calculated using the disparity map as shown in equations (4.6) and (4.7).

$$E_{LR}(x) = d_L(x+1) - d_L(x) \tag{4.6}$$

$$E_{RL}(x) = d_R(x) - d_R(x+1) \tag{4.7}$$

The resulting error values E_{LR} and E_{RL} are compared with a certain threshold, which has been chosen empirically, to determine whether they are occlusion boundaries or not as a first step in the left and right images, respectively. As the occlusion boundaries are detected, it is necessary to fill the occlusion region itself. The width of the occlusion regions that need filling is determined from the difference in the disparities at the occlusion border. OCC constraint detects the occlusion areas more accurately (less noise) than the LRC and ORD approaches, since OCC does not depend on the intensity matching. In particular, the detection process of the

occlusion boundaries is accurate but the filling process in the occlusion regions is erroneous, especially in detailed areas. Unfortunately, OCC fails on images that have varying depths in detailed areas (for example, a scene of leaves of a tree) since the detection process is totally dependent on the disparity map.

4.3 Newly-Exposed/Occlusion Detection Algorithm

We propose a novel method for occlusion/newly-exposed area detection. Geometrically, the stereo images result from projecting the same 3D scene in the physical world from a different position. Therefore, the image objects appear to be displaced between the stereo pair, leaving a covered/uncovered area that forms an occlusion/newly-exposed area. Depending on this fact, we implement a novel algorithm to detect the occlusion/newly exposed area based on the physical displacement of the objects between the stereo pair after forward-projecting the sampling space of one image into the other. A block diagram for our proposed algorithm is depicted in Figure 4.6. The detection algorithm is applied on rectified images, so the displacement will be tested just in the horizontal coordinate.

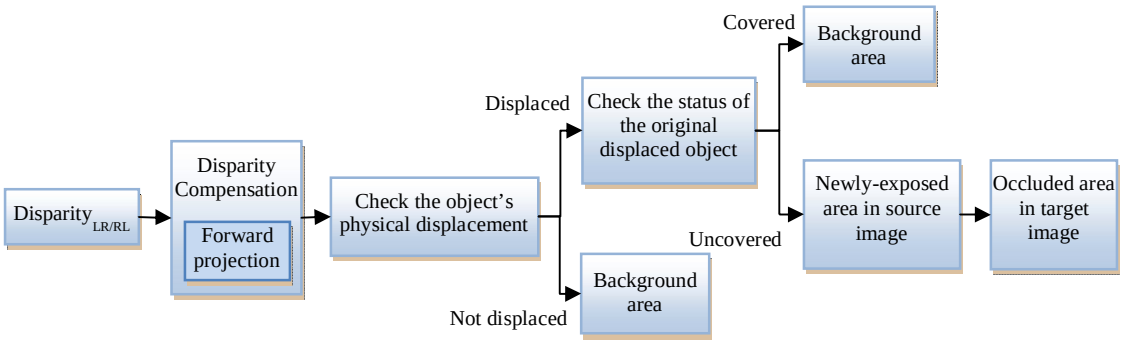


Figure 4.6: A block diagram that illustrate the steps which are needed to detect the newly-exposed area

Practically, the proposed algorithm detects the newly-exposed area in the source image by forward-projecting the sampling space of the source image to the sampling space of the target image by disparity compensation. As a result, a new space is built which contains the new projected points (*PP*). The projected points matrix give us the destination of each pixel in the source image that is physically displaced either

from left to right PP_{LR} or from right to left PP_{RL} , as illustrated in equations (4.8) and (4.9), respectively.

$$PP_{RL}(x) = x + d_R(x), \quad x \in S_R \tag{4.8}$$

$$PP_{LR}(x) = x - d_L(x), \quad x \in S_L \tag{4.9}$$

where x is the spatial position in either left or right image, d_R is the disparity from right to left, d_L is the disparity from left to right, and S_L *and* S_R are the sampling grid space of the left and right image, respectively.

Because some points in the *PP* matrix reserve the same position for the same object in both the source and target images, we have to check whether the image points are physically displaced from the source image to another position in the target image. Practically, if it is in the same position (i.e. disparity is zero), there is no possibility of it being a newly-exposed/occluded area and it is considered as background. On the other hand, if it is physically displaced, there is a possibility of it being occluded. This is defined in equation (4.10),

$$DP(x) = \begin{cases} 0, & PP(x) = x \\ 1, & PP(x) \neq x \end{cases} \tag{4.10}$$

where the *PP* matrix is refer either to the PP_{LR} or PP_{RL} and the DP matrix is defined as the displaced points matrix. The *DP* matrix contains either 1 or 0, where 0 means that the value of x position in the *PP* matrix is equal to its position, and 1 means it is not equal and physically displaced. In the former case, it does not need further processing and it is considered as a background pixel (not occluded). In the latter case, its position will either be covered by another image point or it will be left uncovered. If it is covered, this position will be not considered a newly-exposed area, while if it is uncovered, this is a strong clue to it being a newly-exposed area in the source image (and occluded area in the target image). We will check the original position of the displaced pixel x to see whether it is evacuated by another pixel or not by carrying out a search around it, as in equation (4.11) and (4.12). This search

process is carried out from both sides of the displaced pixel *x* with distance *z* that is equal to the maximum disparity in each row *y*, as illustrated in Figure 4.7.

$$NE_i(x) = \begin{cases} 1, & PP(x + w_i) \neq x \\ 0, & PP(x + w_i) = x \end{cases} \qquad w_i \in [-z, z],\ x \in S_{L/R} \tag{4.11}$$

$$z = \max(d_{L/R}(y)), \qquad y \in S_{L/R} \tag{4.12}$$

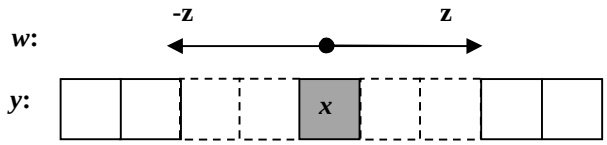


Figure 4.7: Search process around the displaced pixel x

Where w_i is the ith value from *x* in the same row, and $d_{L/R}$ refers either to the left disparity or to the right. As a result, the newly-exposed areas are stored in a new binary image *NE*, where the newly-exposed areas are 1 and the non-newly-exposed areas are 0. If the *NE* is 1, this means that this position in the sampling grid will not be covered by another pixel and can be considered a newly-exposed area, while if *NE* is 0, this means there is another pixel that will cover this place and it can be considered a background area. Visibly, this newly-exposed area in the source image is considered an occluded area in the target image.

4.4 Results and Analysis

An evaluation of our proposed occlusion algorithm is presented in this section. Based on the used material and performed experiments, we will discuss the results, performed measurements and a comparison of the results of our proposed algorithm with three well-known algorithms in this field in the subsequent sections.

4.4.1 Material

To demonstrate the performance of the proposed algorithm, different types of images are tested; these have different levels of occlusion from the simplest (synthetic) to the more complicated (real images). According to the synthetic datasets, one of them

is generated by graphics software and the other is downloaded from Alberta University. On the other hand, two of the real datasets (Teddy and Map) are downloaded from the Middlebury site whereas the Tsukuba dataset belongs to Tsukuba University.

4.4.2 Hardware and Software

The proposed NEOD algorithm is implemented using MATLAB version 7.8, and all the experiments are tested on Intel Core 2 Duo CPU 2.00 GHz 2.00 GHz, 3.00 MB of RAM laptop running under Windows Vista, Home Premium.

4.4.3 Procedure

The proposed algorithm is compared to three well-known algorithms, LRC algorithm, ORD constraint and OCC constraint, under four datasets to demonstrate its performance. All of these approaches are implemented and evaluated against the ground truth data. The availability of ground truth data gives us a good opportunity to compare the proposed occlusion detection results to the ground truth occlusion area qualitatively and quantitatively.

4.4.4 Quantitative Metric

The quantitative comparison with the other detection techniques is accomplished by calculating the number of corresponding matched pixels in the ground truth and the algorithm's detected areas. Hence, four metrics are calculated for each detection technique and the ground truth area. Firstly, true positive *tp* metric represents the number of pixels that are truly classified as occluded areas. The second metric is false negative *fn* which represents the number of background pixels (non-occluded pixels) that are falsely classified as pixels. Thirdly, false positive *fp* is the number of pixels that failed to be classified as occluded area. Fourthly, true negative *tn* is the number of pixels that are truly classified as background pixels.

Depending on these four metrics, *sensitivity* and *specificity* terms are computed to find the probability of each detection algorithm correctly detecting the occlusion area and correctly detecting the background area, respectively.

$$sensitivity = \frac{tp}{tp + fn} \tag{4.13}$$

$$specificity = \frac{tn}{tn + fp} \tag{4.14}$$

The frequently used *Accuracy* metric combines the sensitivity and specificity metrics into one single metric, which is better for comparison.

$$Accuracy = \frac{tp + tn}{tp + fn + fp + tn} \tag{4.15}$$

Unfortunately, the large ratio between the background and the occlusion area does not give a reasonable comparison with just these metrics. To clarify this, let us assume the ground truth occluded area is A1, the tested occluded area is A2, and that this area is contained in the image size 100 × 100. Consider the number of pixels in A1 to be 200; if the test area A2 detects only 40 of these pixels then the accuracy metric for A2 is 0.98, as illustrated in equation (4.16). This level of accuracy is due to the large number of true negative background pixels, which are considered less informative and to be affected by the size of the image

$$Accuracy_{A2} = \frac{40 + 9800}{40 + 160 + 0 + 9800} = 0.984 \tag{4.16}$$

Alternatively, the *Error* metric is used to measure the distortion precision in the occluded result. As the false positive and the false negative gauge the pixel deviation from the occluded area, the *Error* metric, which measures the probability of false positive and false negative classification, is expressed as in equation (4.17).

$$Error = \frac{fp + fn}{tp + fn + fp + tn} \tag{4.17}$$

4.4.5 Experimental Results

4.4.5.1 Synthetic datasets

The first synthetic dataset is chosen to be simple, and it has interleaved blocks with variant depths and static background as shown in Figure 4.8 (a-d). If we use the left image and the disparity from left to right as an input to our algorithm, the obtained result will be the newly exposed area in the left image that is considered as occlusion area in the right image; this area is shown in Figure 4.8 (f) in grey. On the other hand, Figure 4.8 (f) shows the resulting area if the right image and the disparity from right to left are used as an input; this area is coloured black. As we show, the results of the proposed algorithm are accurate and consistent under these synthetic images and they are identical to the ground truth.

The results of LRC, ORD, and OCC constraints are also depicted in Figure 4.8 (g), (h) and (i), respectively. As we note, the results of OCC constraint are close to the proposed algorithm as well as to the ground truth, while the LRC and ORD give a fragmented (not fully detected) occlusion area. However, the ORD constraint detects occluded pixels more than LRC. As we note, a few pixels from the grey rectangle that appear in the ground truth area in Figure 4.8 (e) are detected by LRC and ORD constraints, as shown in Figure8 (g) and (h). This is because the LRC and ORD use the intensity matching for the occlusion detection, and this occluded area is surrounded by three different intensities, which are the two blocks and the background. Also, they fail to detect this area due to its small size.

The demonstration of the shape of the occlusion and the newly exposed areas is depicted in Figure 4.9, where the resulting opposite L shape is due to the presence of two different disparities under the foreground block. As the underlying block (i.e. distant block) is shifted 10 pixels and the foreground block is shifted 20 pixels while the background is static with 0 disparity, a different size for the occlusion area will be obtained. From the side of the underlying block the occlusion area will be 10 pixels in width while it will be 20 pixels in width from the background side.

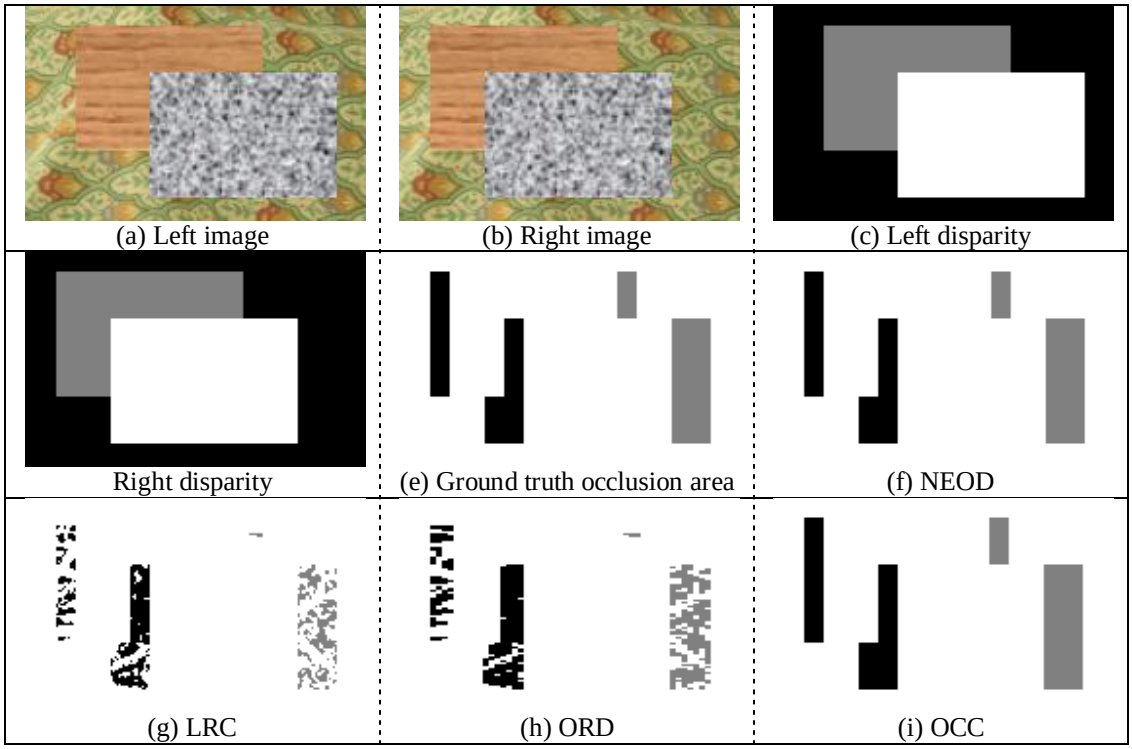


Figure 4.8: Occlusion (black) and newly-exposed areas (grey) on the Two-interleaved synthetic data. (a-d) the original images and their ground truth disparity (e) ground truth occlusion and newly-exposed areas (f) NEOD results (g) LRC results (h) ORD results (i) OCC results

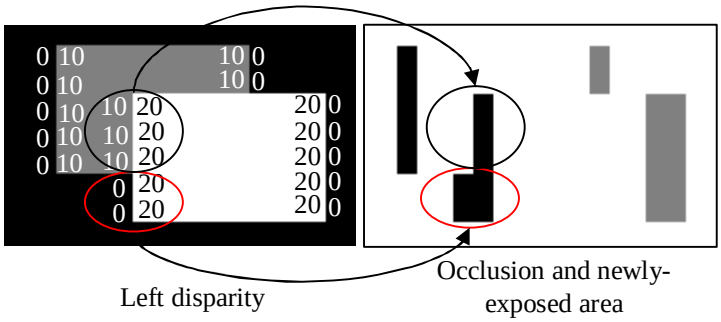


Figure 4.9: Occlusion and newly-exposed area demonstration of the Figure 4.8

The quantitative results obtained from comparing the ground truth of the interleaved-block dataset to the detected occlusion area from LRC, ORD and OCC algorithms and the proposed algorithm are depicted in Table 4.2. As we note, the area detected by the proposed algorithm is identical to the ground truth data, so the sensitivity, specificity and accuracy are 1 while the error is 0. According to the LRC, all the background pixels are truly classified as a non-occluded area; consequently the *fn*

will be 0 and *tn* will be large, generating a sensitivity of 1[4], which is a consequence of using static background.

Table 4.2: Shows the sensitivity, specificity, accuracy, and error values of the two-interleaved synthetic dataset under different four algorithms

Selected threshold		LRC	ORD	OCC	NEOD
1	**Sensitivity**	1	0.9134	0.9711	1
	Specificity	0.9016	0.9278	0.9946	1
	Accuracy	0.9081	0.9263	0.9909	1
	Error	0.0919	0.0737	0.0091	0
5	**Sensitivity**	1	0.9128	0.9711	1
	Specificity	0.8948	0.9211	0.9946	1
	Accuracy	0.9010	0.9203	0.9909	1
	Error	0.0990	0.0797	0.0091	0
10	**Sensitivity**	1	0.9121	0.9711	1
	Specificity	0.8880	0.9117	0.9946	1
	Accuracy	0.8938	0.9117	0.9909	1
	Error	0.1062	0.0883	0.0091	0
15	**Sensitivity**	1	0.9115	0.9711	1
	Specificity	0.8817	0.9033	0.9946	1
	Accuracy	0.8870	0.9039	0.9909	1
	Error	0.1130	0.0961	0.0091	0
20	**Sensitivity**	1	0.9092	0.9711	1
	Specificity	0.8768	0.8955	0.9946	1
	Accuracy	0.8817	0.8964	0.9909	1
	Error	0.1183	0.1036	0.0091	0

According to the ORD and OCC, all the background pixels seem truly classified as non-occluded, but it is clearly shown in Figure 4.8 (h) and (i) that the occlusion area is wider than the ground truth, so a few pixels are classified as *fn*. However, the number of *fn* pixels using OCC constraint is fewer than those using ORD constraint. As we note from the results of LRC and ORD algorithms, the smaller threshold increases the number of *tp* and decreases the *fp*, which in turn gives a higher specificity and accuracy with fewer errors. According to the OCC constraint, the results are very close to the ground truth due to independency in the intensity matching; the small difference between them can be traced to the difference in the widths of the occlusion areas, as we mentioned above. Using this dataset, the

[4] This is because the sensitivity is determined by dividing the *tp* by the summation of *tp* and *fn;* if the *fn* is 0 then the sensitivity will be the division of *tp* by *tp* which will result in 1 regardless of the value of *tp*.

occlusion results are steady under different thresholds, where the range of the threshold values of OCC constraint is smaller than LRC and ORD; they are located between 1 and 5.

The second synthetic dataset is chosen from the Alberta University datasets and it is chosen because it is more complicated than the two interleaved datasets. The original image and its disparity map are shown in Figure 4.10 (a) and (b), respectively. The results of the proposed algorithm and LRC, ORD and OCC algorithms are depicted in Figure 4.10 (d), (f), (g), and (h), respectively. As we note, the proposed algorithm and the OCC constraint are very close to the ground truth results, as shown in Figure 4.10 (c), while the LRC and ORD detect the occlusion area but with a very noisy background.

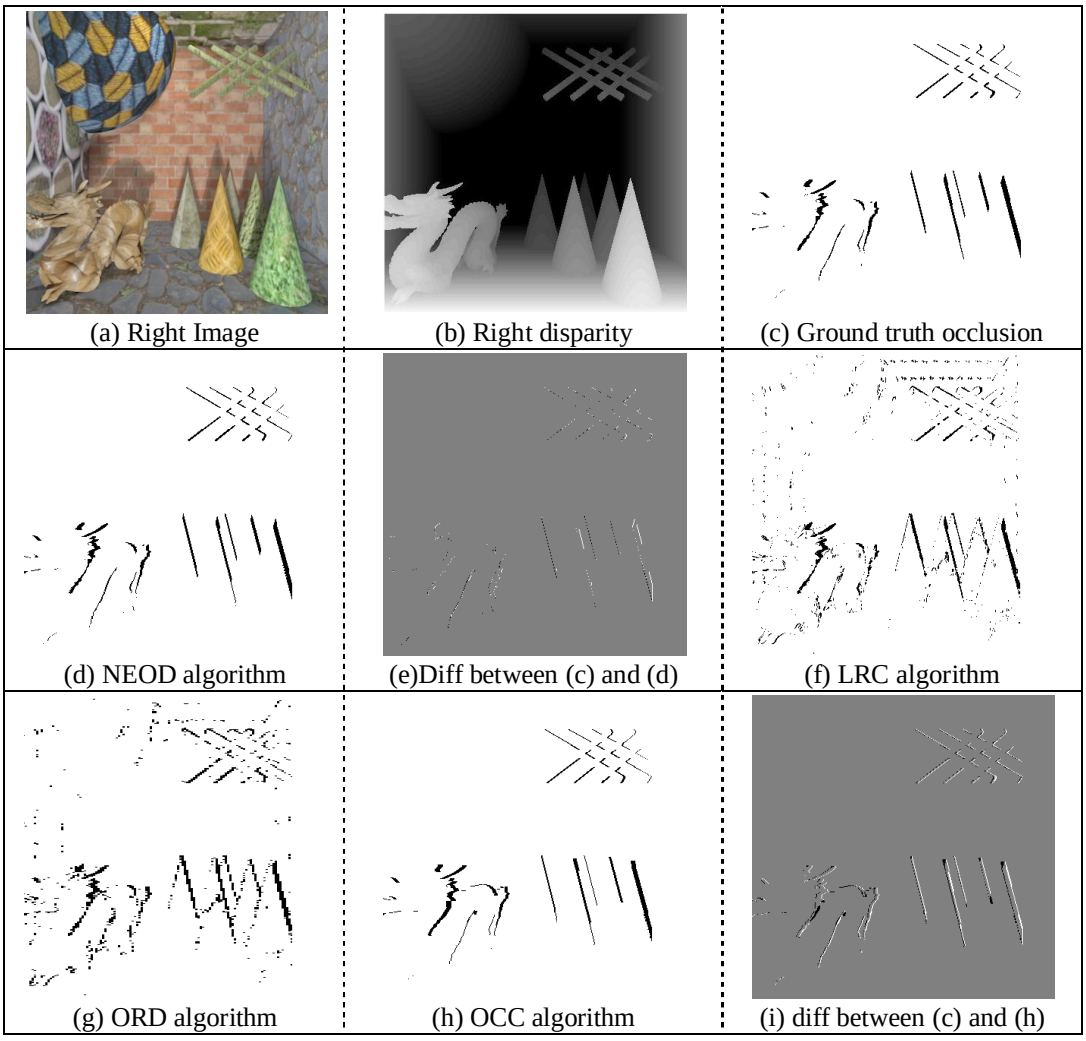


Figure 4.10: Occlusion areas on one of the Alberta University synthetic datasets under different algorithms.

The quantitative results generated from comparing the ground truth data to the resulting occlusion areas with four different algorithms is depicted in Table 4.3. As we note, under small threshold values for LRC, ORD and OCC algorithms, many pixels are classified falsely as occluded; consequently the number of *fn* pixels is large and the number of *fp* pixels is small, thus generating low sensitivity and high specificity values respectively. In this case, a larger threshold is chosen to decrease the *fn* which in turn increases the *tn*. The range of the threshold values for OCC constraint is smaller than for ORD and LRC; they are between 2 and 12 with step 2.

Table 4.3: Shows the sensitivity, specificity, accuracy, and error values of the Alberta University synthetic dataset under four different algorithms

Selected threshold		LRC	ORD	OCC	NEOD
5	**Sensitivity**	0.2118	0.1032	0.6494	0.9249
	Specificity	0.9973	0.9973	0.9854	0.9981
	Accuracy	0.9313	0.8410	0.9824	0.9967
	Error	0.0687	0.1590	0.0176	0.0033
8	**Sensitivity**	0.3662	0.1766	0.6657	0.9249
	Specificity	0.9964	0.9964	0.9882	0.9981
	Accuracy	0.9675	0.9174	0.9840	0.9967
	Error	0.0326	0.0826	0.0160	0.0033
12	**Sensitivity**	0.5593	0.2917	0.6970	0.9249
	Specificity	0.9954	0.9956	0.9928	0.9981
	Accuracy	0.9831	0.9568	0.9872	0.9967
	Error	0.0169	0.0432	0.0128	0.0033
15	**Sensitivity**	0.7291	0.4263	0.7001	0.9249
	Specificity	0.9946	0.9948	0.9931	0.9981
	Accuracy	0.9892	0.9745	0.9875	0.9967
	Error	0.0108	0.0255	0.0125	0.0033
18	**Sensitivity**	0.8171	0.5147	0.7001	0.9249
	Specificity	0.9936	0.9939	0.9931	0.9981
	Accuracy	0.9906	0.9806	0.9875	0.9967
	Error	0.0094	0.0194	0.0125	0.0033

As we note, the ORD produces a higher number of *fn* pixels that consequently give a lower sensitivity and accuracy. The OCC results seems to be free from the background noise but, in reality, not all the occlusion area is located in the true position, so the number of *tp* and fn is relatively high, consequently giving a low sensitivity. This is proved in Figure 4.10 (i) which shows a high difference between the ground truth data and the OCC results at the occlusion border. A high specificity value is returned to the high number of *tn* pixels. According to the proposed

algorithm results, it is clear from the occlusion results in Figure 4.10 (d) and the difference from the ground truth data in Figure 4.10 (e) that a small error is generated using our proposed algorithm due to the sub-pixel disparity. In addition, the proposed algorithm gives a high sensitivity, specificity, and accuracy with fewer errors than the other algorithms.

4.4.5.2 Real datasets

The first real dataset analysis is performed on the University of Tsukuba dataset which contains the image sequence and ground truth for disparity and occlusion area shown in Figure 4.11.(a), (b), and (c), respectively.

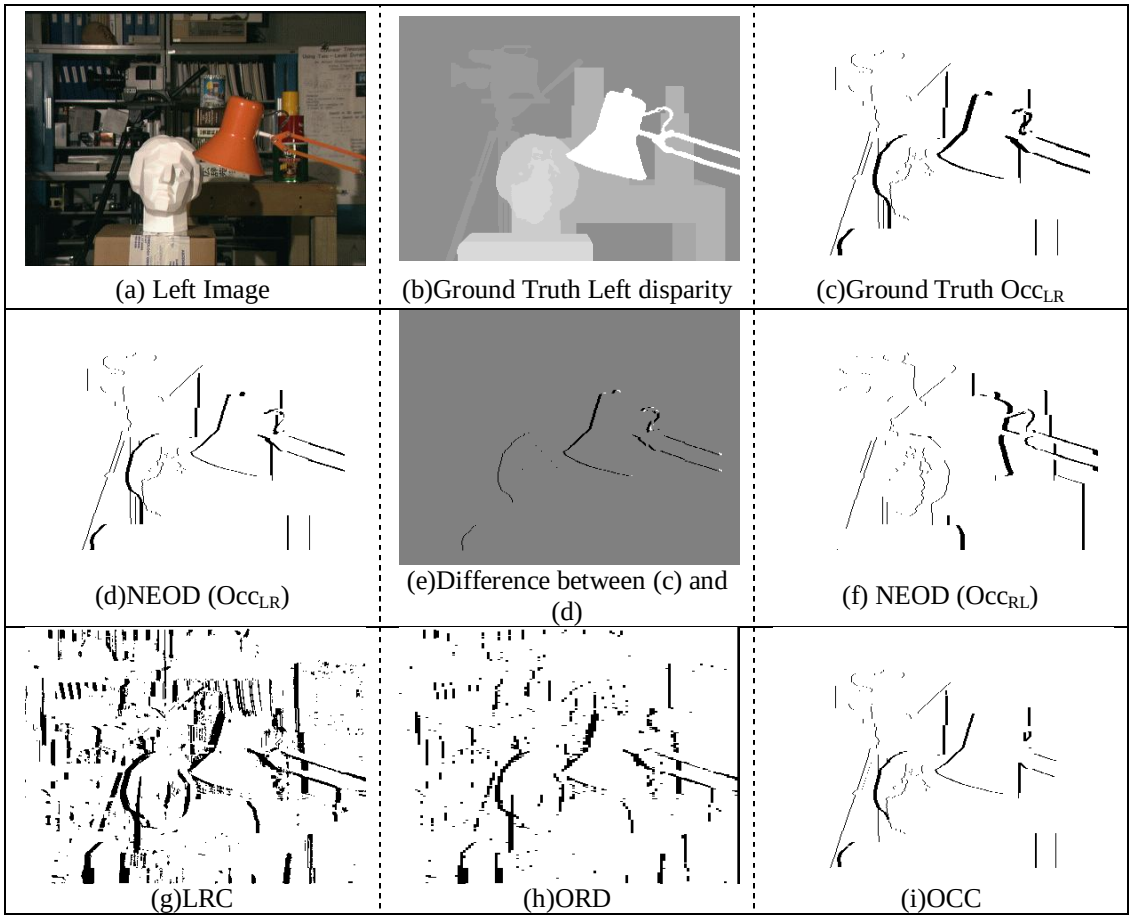


Figure 4.11: Novel algorithm (NEOD) results on the real images (Tsukuba) from both directions where the ground truth of occlusion is available and they are compared to the LRC, ORD and OCC algorithms

The sensitivity, specificity, accuracy and error values for each algorithm are shown in Table 4.4 under different thresholds. The reason for generating a low sensitivity

and high specificity for LRC and ORD algorithms is the same as that illustrated in the previous example of the Alberta dataset. However, the sensitivity of Tsukuba is much lower than the sensitivity of Alberta due to the high number of *fn* pixels in the background, even with a large threshold. The range of the threshold values for OCC constraint is smaller than ORD and LRC where they are divided by 10 to give threshold values between 1 and 5. However, sensitivity, specificity, accuracy and error values for the novel algorithm are steady under different thresholds because it does not use any threshold.

As we show in Figure 4.11, the proposed algorithm detects all of the occlusion area as in the ground truth image for the Tsukuba image, but some edges appear to be thinner due to the sub-pixel accuracy, as shown in Figure 4.11 (d). The occlusion area from the other direction is shown in Figure 4.11 (e).

Table 4.4: Shows the sensitivity, specificity, accuracy and error values of the Tsukuba dataset under different four algorithms

Selected threshold		LRC	ORD	OCC	NEOD
10	**Sensitivity**	0.0718	0.0898	---	0.9813
	Specificity	0.9866	0.9841	0.9743	0.9941
	Accuracy	0.7938	0.8652	0.9743	0.9939
	Error	0.2062	0.1348	0.0257	0.0061
20	**Sensitivity**	0.0862	0.1077	0.7162	0.9813
	Specificity	0.9849	0.9823	0.9775	0.9941
	Accuracy	0.8509	0.9043	0.9763	0.9939
	Error	0.1491	0.0957	0.0237	0.0061
30	**Sensitivity**	0.0823	0.0914	0.7161	0.9813
	Specificity	0.9816	0.9789	0.9783	0.9941
	Accuracy	0.8792	0.9207	0.9768	0.9939
	Error	0.1208	0.0793	0.0232	0.0061
40	**Sensitivity**	0.0810	0.0798	0.8722	0.9813
	Specificity	0.9798	0.9772	0.9856	0.9941
	Accuracy	0.8986	0.9310	0.9841	0.9939
	Error	0.1014	0.0690	0.0159	0.0061
50	**Sensitivity**	0.0783	0.0692	0.8930	0.9813
	Specificity	0.9784	0.9762	0.9881	0.9941
	Accuracy	0.9124	0.9386	0.9866	0.9939
	Error	0.0876	0.0614	0.0134	0.0061

The other two datasets, Map and Teddy, are downloaded from the Middlebury site. They are chosen because they vary in their texture complexity; the former is a very simple real image example as shown in Figure 4.13 (a), while the latter is very

complicated as in Figure 4.12 (a). Comparison of the four algorithms is performed on the Teddy dataset as shown in Table 4.5. Generally, the occlusion results of LRC and ORD are similar in their detection accuracy, in that they correctly detect the occlusion area but are fragmented. In addition, LRC and ORD are both sensitive to the noise, since they depend on the intensity matching error as an indicator of the occlusion area. Numerically, there is a variation in the accuracy metric between LRC and ORD on the Teddy image and Tsukuba, where the accuracy for the ORD is higher since it produces fewer errors than LRC on the Tsukuba image, and vice versa on the Teddy image. Under the Map image, LRC gives similar accuracy to the Tsukuba image. This is clearly shown on the Tsukuba, Teddy, and Map images in Figure 4.11 (g, h), Figure 4.12 (d, e), and Figure 4.13 (d, e), respectively.

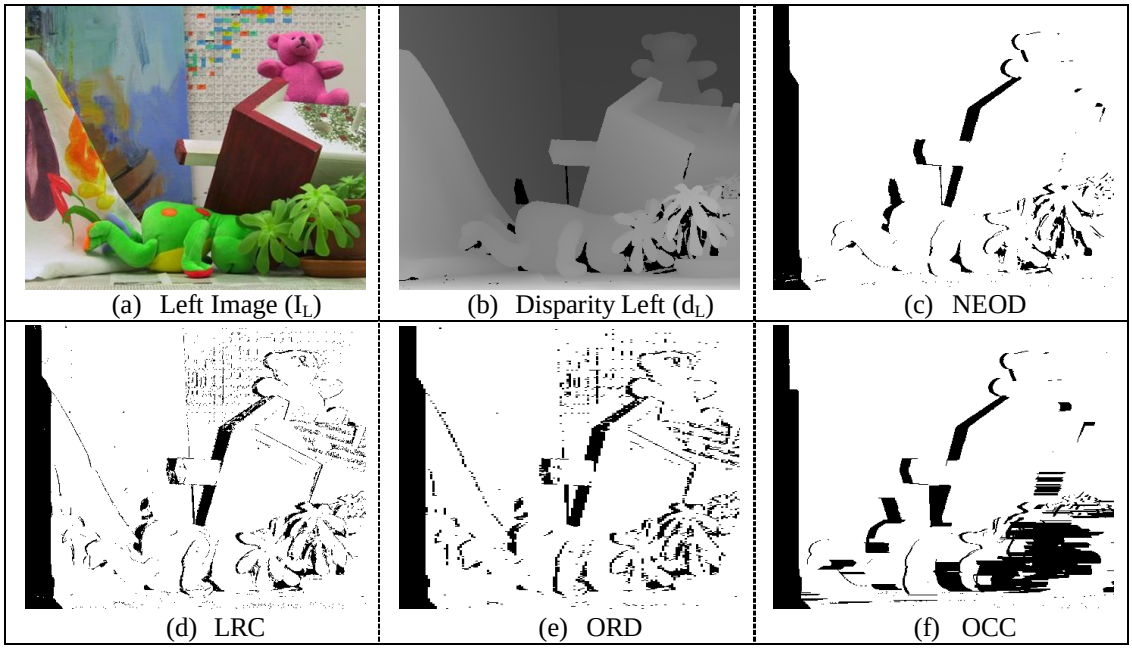


Figure 4.12: Occlusion detection results of the three well-known algorithms and the proposed one on the Teddy image.

On the other hand, the OCC constraint gives different detection results under different types of image textures where, in the case of a simple texture image, it gives accurate results as shown in Figure 4.8 (i), Figure 4.10 (h), Figure 4.11 (i) and Figure 4.13 (f), while a corrupted result is obtained in the case of complex texture images, as shown in Figure 4.12 (f). In particular, the occlusion detection result in the Teddy image is accurate in some places in the image and is corrupted in other

places that contain a detailed and fine texture (varying depths), especially close to the leaves. This is because the OCC constraint detects the occlusion boundary as the first step, as such an area makes a jump close to the boundary, then fills from this boundary by the difference in disparity at this jump. Since the OCC constraint depends only on the disparity map to find the occlusion area, the result is less sensitive to the noise.

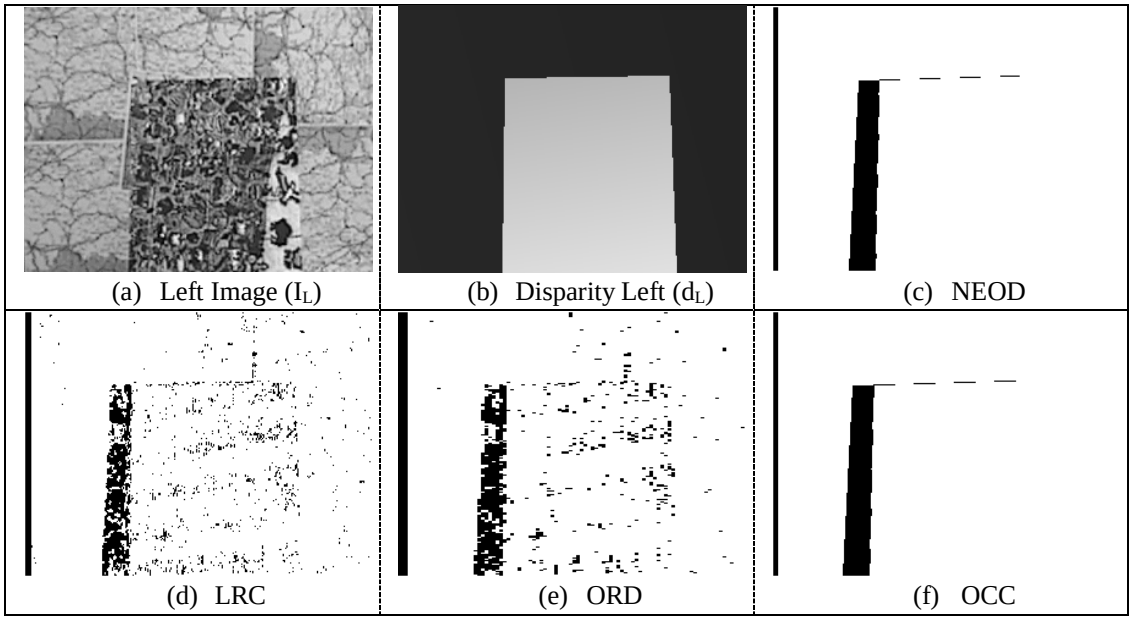


Figure 4.13: Occlusion detection results of the three well-known algorithms and of the proposed one on the Map image.

The detection results of the proposed algorithm are superior to the LRC, ORD, and OCC algorithms under different types of image textures since it yields a higher accuracy and lower error precision at the same time. Under the Tsukuba dataset, the accuracy of 0.9939 is considered a good detection rate comparing to the other algorithms, which fall between 0.7938 and 0.9124 under different threshold values. In experiments under a simple texture image, OCC constraint gives results comparable to those of the proposed algorithm, but fails under the complicated one, as mentioned above.

Since LRC, ORD, and OCC depend on computing matching error, they need to select a threshold for comparison in the detection process. Unfortunately, this selection process is very sensitive and frustrating since, under the same algorithm,

different thresholds are selected for different images. In this chapter, the threshold values for each algorithm (LRC, ORC, and OCC) are determined empirically for different images. On the other hand, the proposed algorithm detects the occlusion area without using any threshold, because it does not depend on any intensity matching errors with which to compare a threshold. According to the input parameters, LRC and ORD need the left image, right image and the disparity map as inputs to detect the occlusion area from one direction, while OCC and the proposed algorithm just need the disparity map from one direction to detect the occlusion area.

Table 4.5: Shows the sensitivity, specificity, accuracy, and error values of the Teddy dataset under four different algorithms

Selected threshold		LRC	ORD	OCC	NEOD
10	**Sensitivity**	0.5825	0.3921	0.5105	0.9799
	Specificity	0.9886	0.9933	0.9641	0.9826
	Accuracy	0.9078	0.8087	0.8789	0.9823
	Error	0.0922	0.1913	0.1211	0.0177
15	**Sensitivity**	0.7539	0.5837	0.5117	0.9799
	Specificity	0.9845	0.9896	0.9642	0.9826
	Accuracy	0.9503	0.9084	0.8794	0.9823
	Error	0.0497	0.0916	0.1206	0.0177
20	**Sensitivity**	0.8369	0.6868	0.5138	0.9799
	Specificity	0.9807	0.9859	0.9642	0.9826
	Accuracy	0.9621	0.9366	0.8801	0.9823
	Error	0.0379	0.0634	0.1199	0.0177
25	**Sensitivity**	0.8835	0.7495	0.5167	0.9799
	Specificity	0.9764	0.9820	0.9642	0.9826
	Accuracy	0.9655	0.9480	0.8812	0.9823
	Error	0.0345	0.0520	0.1188	0.0177
30	**Sensitivity**	0.9108	0.7893	0.5190	0.9799
	Specificity	0.9719	0.9776	0.9641	0.9826
	Accuracy	0.9652	0.9524	0.8820	0.9823
	Error	0.0348	0.0476	0.1180	0.0177

Another experiment is performed on the Teddy dataset under inaccurate disparity, i.e. non-ground truth disparity, to show the accuracy of the proposed algorithm. The estimated disparity is calculated using a region-based stereo matching algorithm using cooperative optimization [127]. As we note in Figure 4.14 (b), the estimated disparity does not have black areas which represent areas of unknown disparity; consequently the occlusion results do not include those areas in the occlusion

detection result with the ground truth disparity, as shown in Figure 4.14 (c). Thus, the unknown disparity areas are marked and added to the occlusion result under estimated disparity in order to obtain the differences between the occlusion detection results under accurate disparity. As shown in Figure 4.12 (d) and (e), a slight difference is noticed between those images.

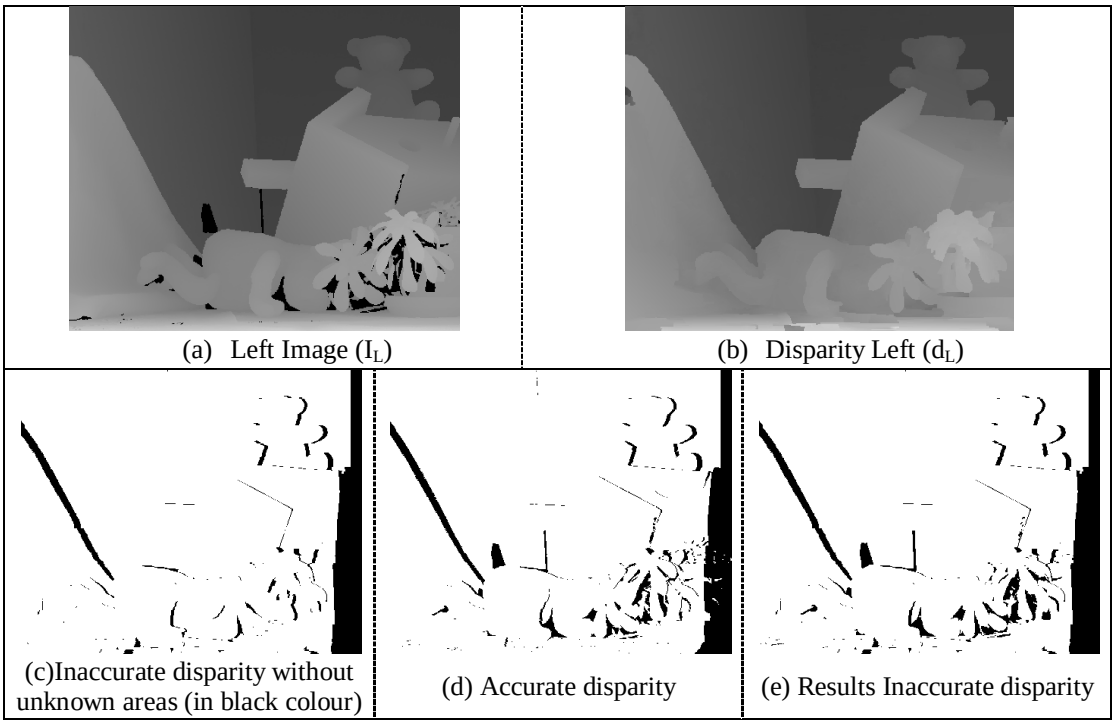


Figure 4.14: Occlusion detection results under accurate and inaccurate disparity on the Teddy image.

4.5 Summary

In this chapter, a novel occlusion detection algorithm has been proposed to overcome the deficiencies of the previous occlusion detection algorithms. The performance of the proposed algorithm is tested under 2 synthetic datasets and 3 real datasets. We have compared our algorithm results with three well-known occlusion detection algorithms (LRC, ORD, and OCC) qualitatively and quantitatively, and showed that the proposed algorithm results outperform these algorithms under both types of dataset. Our algorithm has many advantages; for example, the occlusion area is detected without any fragmentation, the results are consistent under different types of images and matching algorithms, it needs few input parameters, and it is easy to implement. We performed various analyses and computer simulations which

confirmed that the proposed algorithm detects the occlusion areas with a high accuracy (0.9939 and .9823) and a low number of errors (0.0061 and 0.0177) for well-known datasets, Tsukuba and Teddy. This computer simulation also confirmed that the proposed algorithm detects the occlusion for two synthetic datasets with a high accuracy (1, 0.9967) and low number of errors (0, 0.0033).

CHAPTER

5

Block-based Inpainting Technique for Occlusion Area (BITO)

Since the structure of the occluded area is very important for the reconstruction process, a reliable inpainting technique (disparity extrapolation) is required to recover the occluded area. Therefore, in this chapter we propose a novel disparity inpainting algorithm for the occlusion areas, after detecting their location using the proposed algorithm explained in chapter 4.

A Block-based Inpainting technique is proposed in this chapter for occlusion areas with continuation of the edge discontinuities; hereafter, we will call it BITO. This algorithm uses a variable block size to inpaint the occlusion area; the block size depends on the existing texture within each block. Our technique inpaints the occluded area accurately while preserving the object edges determined from one of the stereo images, since occluded area intensities are known in one of them. This idea is inspired by the image-driven disparity inpainting technique [93] that inpaints the disparity holes using Bertalmio et al's inpainting approach [92], although it differs in the diffusion process into the occlusion area where it is guided by one of the stereo image gradients.

We compare the proposed inpainting technique to the state-of-of-the-art occlusion-filling and inpainting techniques qualitatively and quantitatively.

5.1 Introduction

Geometrically, the stereo images result from projecting the same 3D scene in the physical world from different positions. So, the image objects appear to be displaced between the stereo pair leaving a covered/uncovered area that forms an occlusion/newly-exposed area. Occluded areas have an impact on two problems: disparity estimation and intermediate view reconstruction. The intensities of those areas are very useful in intermediate view reconstruction, practically; if there is no information about the structure of the occluded area, the reconstruction process cannot reconstruct these areas. Therefore, inpainting the disparity in the occlusion area is necessary; this is also called occlusion handling or disparity extrapolation.

Originally, image inpainting was used to remove a certain object from the image or to repair a corrupted image, i.e. image restoration. This is accomplished by applying a mask on a certain object or the corrupted area on the image that needs inpainting. In occlusion filling, since the occluded area can be considered as the applied mask on a disparity image, the inpainting techniques are considered a suitable solution to fill the occluded area. In [92], the first digital image inpainting technique was introduced. In due course, several studies were carried out to improve Bertalmio *et al's* inpainting approach [94, 95, 128]. Recently, several researchers have developed inpainting approaches to inpaint the occlusion areas in the disparity images [93, 102] that require special filling techniques. In general, inpainting techniques give good-quality results but are computationally expensive.

In the following sections, we will review the state-of-the-art inpainting techniques, starting with the simplest and moving on to the more complicated techniques.

5.1.1 Horizontal Extrapolation using Depth Information HEDI (Depth Constancy Assumption)

This technique is used to calculate the disparity in the occluded area based on the depth information, and is considered the simplest approach. This technique assumes the depth will be the same within a small area of neighbours and that the holes belong to the background, avoiding the foreground objects since they have been

displaced, leaving a gap behind. This assumption is based on the fact that the closer objects will have a large disparity, while the distant objects, which include the occluded area, will have small disparities [58, 76, 91]. Thus, if there is an occluded point located between two non-occluded points, the occluded point is filled using the smallest non-occluded point, and this is illustrated in the following equation.

$$d(h) = \begin{cases} h_l & if \quad d(h_r) > d(h_l) \\ h_r & if \quad d(h_l) > d(h_r) \end{cases} \tag{5.1}$$

Where d refers to disparity image, h is the hole to be inpainted, and h_l and h_r are the first image points to the left and right of the hole. This assumption is not always valid; it fails if there is a lot of change in the depth of the image. Specifically, it does not take the continuation of the edges into consideration in the region to be filled.

5.1.2 Bertalmio *et al' s* Inpainting Approach

Using this approach, a gap in the image is filled or inpainted by continuing the structure of the surrounding area of the gap [92]. This is done in two steps: the first one continuing the gradient of the surrounded area into the gap; the second one diffusing the available intensities into the gap using anisotropic diffusion, which diffuses the intensities taking into consideration the underlying image gradient. This is done several times until the whole region is inpainted. Specifically, we can consider Ω as the area to be inpainted and ∂Ω as the border of this area, as shown in Figure 5.1. In the first step of inpainting, the gradient at ∂Ω is continued into Ω, preserving its direction. Then, the structure of ∂Ω is diffused into the gap Ω. The inpainting process can be represented by the following equation:

$$I^{n+1}(i, j) = I^{n}(i, j) + \Delta t \; I_t^n(i, j), \qquad \forall (i, j) \in \Omega \tag{5.2}$$

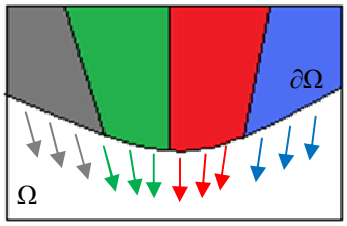
where n is the nth inpainting iteration, Δt is the enhancement rate, *(i, j)* represent the coordinates of the inpainted pixel, and $I_t^n(i, j)$ refers to the enhancement that needs to be added to the image $I^n(i, j)$ to produce $I^{n+1}(i, j)$, as represented in equation (5.3). The inpainting process is continued until $I^n(i, j)$ is equal to $I^{n+1}(i, j)$, or the enhancement rate becomes lower than a certain threshold.

$$I_t^n(i,j) = \overrightarrow{\delta L^n}(i,j).\overrightarrow{N^n}(i,j), \quad (5.3)$$

where $L^n(i, j)$ is the available information that will be diffused, $\overrightarrow{\delta L^n}(i,j)$ quantify how much the information differs from $L^n(i, j)$, and $\overrightarrow{N^n}(i,j)$ determine the diffusion direction. Since the diffusion process should be smooth, $L^n(i, j)$ is expressed by Laplacian equation as in the following:

$$L^n(i,j) = I_{xx}^n(i,j) + I_{yy}^n(i,j), \quad (5.4)$$

Where I_{xx}^n and I_{yy}^n represent the horizontal and vertical second derivatives, respectively. For more details about these equations, please see the original paper [92]. Since this inpainting technique diffuses the structure of the available intensities smoothly without texture replication, a blurred inpainted area is unfortunately produced. This algorithm is considered more complicated than the depth constancy algorithm and it needs more time to inpaint a certain hole as it depends on the iterative filling.



Ω refers to the hole to be inpainted
∂Ω refers to the boundary of the hole

Figure 5.1: Illustrates the diffusion process of Bertalmio *et al's* inpainting approach

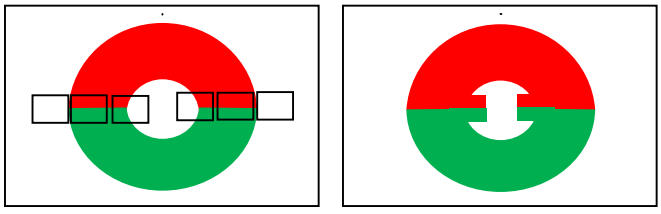
5.1.3 Image-Driven Disparity Inpainting

This algorithm is similar to the Bertalmio *et al's* inpainting algorithm but here the inpainting process is used to inpaint the occlusion areas in the disparity map [93]. Since the underlying images that are used to estimate the disparity map are available, the image and disparity gradient will coincide. Thus, the first step of the Bertalmio *et al's* inpainting algorithm is ignored. In general, this algorithm extrapolates the disparity value in the occluded area by using anisotropic diffusion which is guided by image gradient.

5.1.4 Exemplar-based Inpainting

All the methods described above diffuse the structure of the surrounding area smoothly into the hole without texture replication, which consequently produces a blurred inpainted region. Therefore, many studies have been conducted to combine texture synthesis and inpainting [94-96]. In [95], they processed the image twice: one is processed by texture synthesis and the other by inpainting. Subsequently, the resulting images are blended into one combined output image, but this method still produces blurriness in the inpainted region. Harrison [96] was the first to use the exemplar-based synthesis, where the holes are inpainted based on the level of the texturedness of the surrounding pixels. Although this appears to be a good method, it can be strongly affected by an image noise.

Meanwhile, Criminisi *et al.* [94] combine the advantages of texture synthesis and structure inpainting into one technique. The synthesized pixel value is diffused from the surrounding area in a similar way to the diffusion of pixels in the inpainting process, which results in inpainting both texture and structure. In detail, this technique divides the boundary regions into patches; each patch is centred on the edge of the hole to be assigned a calculated priority, and the patch with highest priority is the first one to be filled. This prioritization takes into consideration the strong edges and the amount of reliable pixels that can help in the filling procedure in each patch. After selecting a certain patch, this technique searches the surrounding area for a similar patch to copy its texture to the target region, as shown in Figure 5.2.



(a) (b)

Figure 5.2: Shows exemplar-based filling procedure.

Many studies were subsequently developed to improve the performance of Criminisi *et al's* inpainting technique [97, 129]. In [129], local consistency in the inpainted

region is preserved by studying the relationship between the selected patches in the boundary area and the similar patches in the neighbourhood. Consequently, image details and sharpness are preserved in the inpainted area. Very recently, another modification was proposed in [97] to fill the occlusion area. These modifications yield a more efficient search process to find similar patches in the neighbourhood, improving the quality of the filled region by modifying the priority assignment procedure to cope with the nature of the occlusion hole. Instead of choosing one similar patch from the candidate patch, they use a weighted non-local mean of k patches to inpaint the disocclusion hole.

5.1.5 Laplacian Filling

The derived kernels from the Laplacian equation in (5.4) can be used as smoothness filters which represent the second derivative of the image in both directions. These filters can be used to fill a certain hole in the image by applying them on the surrounding areas and diffusing them smoothly into the hole. The implementation of this technique is available in ROIFILL MATLAB function which is employed in [98] to inpaint a certain hole in the image.

5.1.6 Fields of Experts (FOE) Inpainting

This technique is based on the fields of experts model which consists of a group of filters learned from a standard database of real images with high diversity [99]. This FOE is a modelling of the extension of the Markov Random Field that can capture the statistics of the real images. The FOE is employed in two areas: image de-noising and image inpainting. According to the image inpainting procedure, the masked area, i.e. hole, is inpainted by exploiting prior FOE. Specifically, a straightforward gradient tracking procedure is used, extracting the local structure characteristics from the response of the pre-trained filters, without using image gradient direction explicitly.

This method preserves the continuation of the edges better than Bertalmio *et al's* method. Unfortunately, the filling procedure is based on intensities diffusion without taking into consideration texture reproduction that in turn generates a blurred area,

which is considered a disadvantage of this method in addition to the computational complexity.

5.1.7 Smoothing of Depth Maps to Avoid Holes

Pre-processing the depth map by smoothing filters has been addressed in the literature to reduce the size of the occluded area or remove it completely due to the vertical design of the occlusion holes. An average filter is employed to inpaint the occluded area in [100], but this generates image artifices which yield a low perceived quality. Pre-processing with a symmetric Gaussian filter is proposed in the same paper to fill the occluded area. Using this filter, the strength of smoothing in vertical and horizontal directions is the same which producing a geometric distortion in the vertical lines that become curved. Therefore, the asymmetric Gaussian filter [101] is proposed to solve this problem by increasing the smoothness in the vertical direction and decreasing it in the horizontal direction. Unfortunately, the distortion in the horizontal direction might still exist. Therefore, another technique is proposed in the literature [130] that smoothes the edges with different filters depending on the strength of the gradient, as a higher gradient in a horizontal direction needs a stronger smoothness.

5.1.8 A Modified-Inpainting Technique for Occlusion Filling

This technique modifies one of the inpainting techniques in the literature to cope with the occlusion filling problem [102]. This is because the digital inpainting techniques fill the holes from all the surrounding areas. On the other hand, the occluded area should be filled from the background information based on the depth constancy assumption described in section 5.1.1. Therefore, all the surrounding areas of the hole are modified to be just from the background, as shown in Figure 5.3. Then, the hole is filled by Telea's inpainting technique [114]. This method is considered computationally expensive.

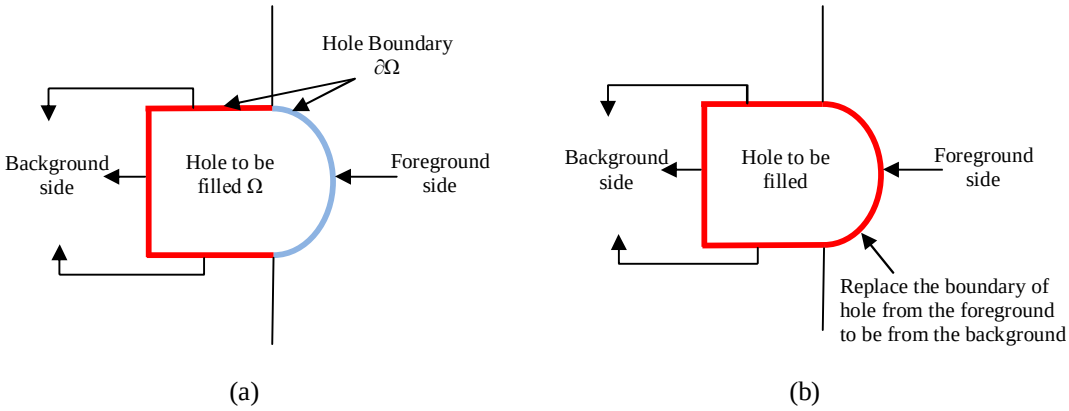


Figure 5.3: Illustrating the general idea of the modified-inpainting technique for occlusion filling

5.1.9 Comparative Studies on Occlusion Filling

A comparative study analysis of six different techniques for filling the disocclusion area is presented in [103]. This study shows that the HEDI technique gives the best results in real time followed by ROIFILL and the variational inpainting [131] techniques, which are computationally expensive.

Recently, another comparative analysis study of three different occlusion filling techniques has been conducted in [104]. The exemplar-based inpainting technique of Criminisi *et al*. [94] is compared to the pre-processed depth map with a specified filter [132] and the computationally efficient inpainting algorithm of Oliveira *et al*.[128]. The study shows that Criminisi *et al's* technique outperforms the other two techniques in terms of the image quality while, in terms of computational complexity, it is considered the most expensive. Consequently, this comparative study encourages the researcher to seek further improvements to the exemplar-based technique.

5.2 Block-based Inpainting Technique for Occlusion Area (BITO)

The proposed Block-based Inpainting Technique for Occlusion area (BITO) is a novel technique for inpainting the occlusion/newly-exposed area based on block-filling with continuation of edge discontinuities. It uses a variable block size that is determined by the existing texture within each block. By assuming that the occluded

area intensities are known in one of the stereo images[5], our technique diffuses object edges in the occluded area by exploiting the available stereo images. This is shown in Figure 5.4, and we notice that the image intensities of the occluded area that appear in Figure 5.4 (b) are available in the stereo image in Figure 5.4 (a). So, we exploit the edge map of the stereo images to employ it in the inpainting algorithm.

The framework of the proposed algorithm is depicted in Figure 5.5, illustrating the pipeline of the inpainting algorithm. The input to the proposed algorithm consists of two images: the reference image and its disparity. The reference image is used to extract the edge map from it, while the occluded areas that need filling exist in the disparity map. The output from inpainting steps will be inpainted disparity without occluded areas.

Practically, our inpainting technique divides the disparity into two parts. The first part is filling the occlusion area in the extreme left- and right-hand sides of the image as shown in Figure 5.6 (a) and Figure 5.6 (b), which will be filled from the foreground object while continuing the edges.

(a) (b)

Figure 5.4: Shows how the intensities of the occluded area are available in the stereo image: (a) the original stereo image (b) Its disparity with occluded area masked in black.

[5] This assumption fails when the image points are occluded in both of the stereo images.

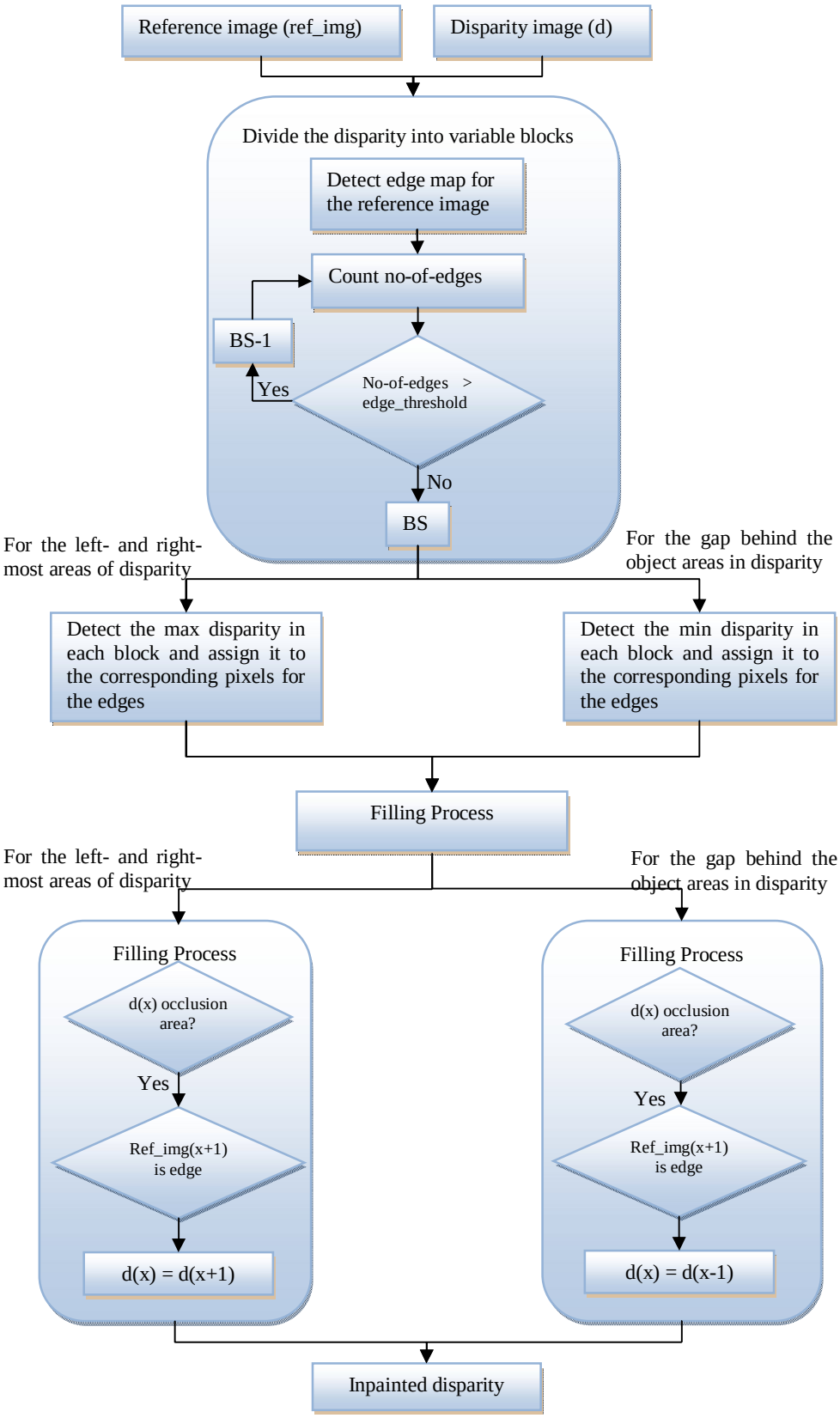


Figure 5.5: Shows the framework of the proposed inpainting algorithm

The filling of the left-most area of the disparity is shown in Figure 5.7 (a), where it clearly needs to be filled from the foreground object. This issue has been ignored in some of the previous works on filling the occlusion area, such as the HEDI technique which fills the occlusion area by assuming that it always belongs to the background. The second part of our technique is to fill the gap behind the object, which is filled from the background while also continuing the edges, as shown in Figure 5.7 (b).

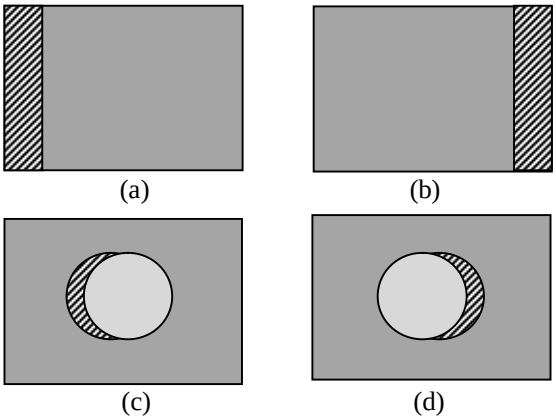


Figure 5.6: Shows the two types of occlusion area (a) and (b) the left- and right-most areas, respectively (c) and (d) the gap behind the object area.

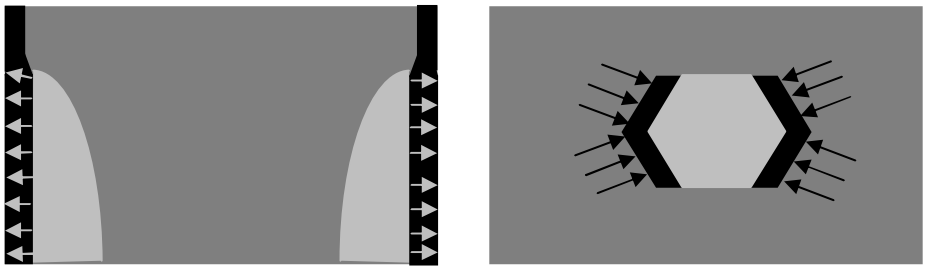


Figure 5.7: Shows the diffusion direction for the two types of occlusion area. (a)The diffusion direction of the left-most and right-most areas is from the foreground object. (b)The diffusion direction for the gap behind the object is from the background area.

In the following section we will explain how the proposed inpainting algorithm fills the occluded area with a disparity step by step:

Step #1: Detect edge map for the reference image, which is detected by Canny edge detection operator as shown in Figure 5.8 (b). As a result a new matrix (*ref_edges*) is

built that contains either 1 or 0, where 1 represents an edge pixel in the reference image and 0 represents non-edge pixel. See equation (5.5).

ref_edges = edge_detection_operator(ref_img) (5.5)

(a) (b)

Figure 5.8: Shows the reference image in (a) and its edge map in (b).

Step #2: Divide the disparity image into blocks starting with large blocks, then start decreasing the block size based on the number of edges within the block until it falls below a predetermined threshold. A large block size could have many different details for different objects. Therefore, we change the size of the block in order to find the appropriate block size that has relatively similar details within it, which helps to inpaint those areas accurately based on objects' texture. The following equations illustrate how each block is divided into smaller blocks.

$$C_{B_i}(x,y) = \begin{cases} 1 & , \quad ref\_edges_{B_i}(x,y) = 1 \\ 0 & , \quad otherwise \end{cases} \tag{5.6}$$

$$no\_of\_edges(B_i) = \sum_{x=1}^{BS}\sum_{y=1}^{BS} C_{B_i}(x,y) \qquad \forall x, y \in Block\,B_i \tag{5.7}$$

$$BS = \begin{cases} BS-1 & , \quad no\_of\_edges(B_i) > edge\_threshold \\ BS & , \quad otherwise \end{cases} \tag{5.8}$$

where *x* and *y* coordinates refer to any pixel inside the block B_i, C_{B_i} is the indicator function that indicates whether the current pixel is an edge or not, *no_of_edges* is the accumulative counter of C_{B_i} that counts the number of edges in each block B_i, *BS* refers to the block size, and *edge_threshold* is a predetermined threshold to determine the minimum number of thresholds that are supposed to be in the block.

According to initial size of the block, we take into consideration the size of the occlusion areas that need filling; if the occlusion areas are large, the initial value for BS is 50, while if they are relatively small, the BS will start by 20. After that, the BS is start decreasing until they have a smallest number of details on it.

Step #3: After determining an appropriate block size in the previous step, we have to find the minimum and maximum disparity in this block depending on the type of occlusion areas that need filling. The maximum disparity in the block is found for the occlusion area located in the extreme left- and right-hand sides of the image, while the minimum disparity is found for areas located behind the objects.

$$min_{B_i} = min\,(B_i) \tag{5.9}$$

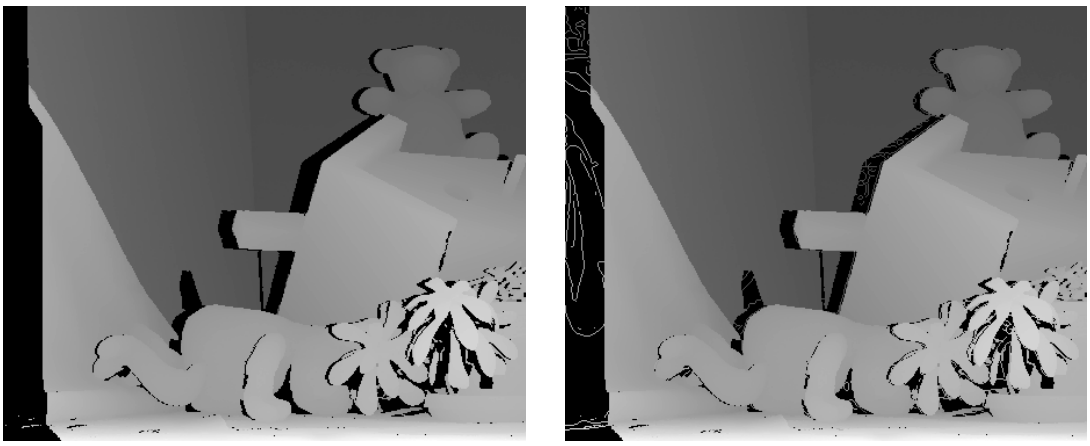
$$max_{B_i} = max\,(B_i) \tag{4.7}$$

Step #4: Assign the corresponding pixels of the edge map in the disparity block by min/max values found in the previous step depending on which type of occluded area in the disparity map we need to fill, as shown in Figure 5.9 (b).

$$d(x) = \begin{cases} min_{B_i} / max_{B_i} & , \quad ref\_edges = 1 \\ 0 & , \quad ref\_edges = 0 \end{cases} \tag{5.10}$$

Step #5: Diffuse the correct disparity inside the occlusion areas based on the type of occluded area: the left-most and right-most areas and the gaps behind the objects. The occluded areas are filled based on the closest guided edges that were filled in the previous step. Specifically, the left-most and right-most areas are filled from right-to-left and from left-to-right, respectively, depending on the direction of the

available intensities. On the other hand, the occlusion areas behind the objects are filled from the direction of the background.



(a) (b)

Figure 5.9: Illustrating step# 4. (a) shows the disparity map with the occluded areas (b) shows filled occluded area with the edge map depending on the type of occluded areas.

5.3 Results and Analysis

An evaluation of our proposed inpainting algorithm is presented in this section. Based on the used material and performed experiments, we will discuss the result analysis, the performed measurements and a comparison of the results for our proposed algorithm with four well-known algorithms in this field in the subsequent sections.

5.3.1 Datasets

The proposed algorithm is tested on four datasets; Teddy, Barn2, and Reindeer datasets are used to illustrate the occlusion filling technique, and the Rocks dataset demonstrates the performance of the unknown area filling. All of these datasets are downloaded from the Middlebury site.

5.3.2 Procedure

The proposed algorithm is compared to four well-known inpainting algorithms, HEDI, RIOFILL, Exemplar-based, and FOE under four datasets to demonstrate its performance. All of these approaches are implemented and evaluated against the

ground truth data. The availability of ground truth data gives us a good opportunity to compare the proposed inpainting technique results to the ground truth data qualitatively and quantitatively. The implementation of the Exemplar-based and FOE are downloaded from the following references [133, 134], respectively, while the ROIFILL is a built-in MATLAB function. Regarding to the HEDI and BITO algorithms, they are our implementation.

5.3.3 Quantitative metric

Peak Signal to Noise Ratio (PSNR) is a metric used to gauge the ratio of the maximum power of a signal to the power of a noise that distorts the quality of the signal. The PSNR is calculated by comparing the inpainted disparity map with the ground truth one to compute the Mean Square Error (MSE) which results from the corrupted noise. The MSE and PSNR are calculated as follows:

$$MSE = \frac{1}{MN}\sum_{x=1}^{N}\sum_{y=1}^{M}[I(x,y) - I'(x,y)]^2 \tag{5.11}$$

$$PSNR = 20.\log_{10}(\frac{\max_p}{\sqrt{MSE}}) \tag{5.12}$$

Where $I(x,y)$ is the ground truth disparity map, $I'(x,y)$ is the inpainted disparity map, M and N are the dimensions of the images and max_p is the maximum pixel value in the image. Higher PSNR refers to higher image quality due to reduced error noise. Unfortunately, the ground truth data contain unknown areas that make the ground truth image different from the inpainted image and are counted as errors in the inpainted image. Therefore, these unknown areas are not taken into consideration through MSE calculation.

5.3.4 Experimental Results

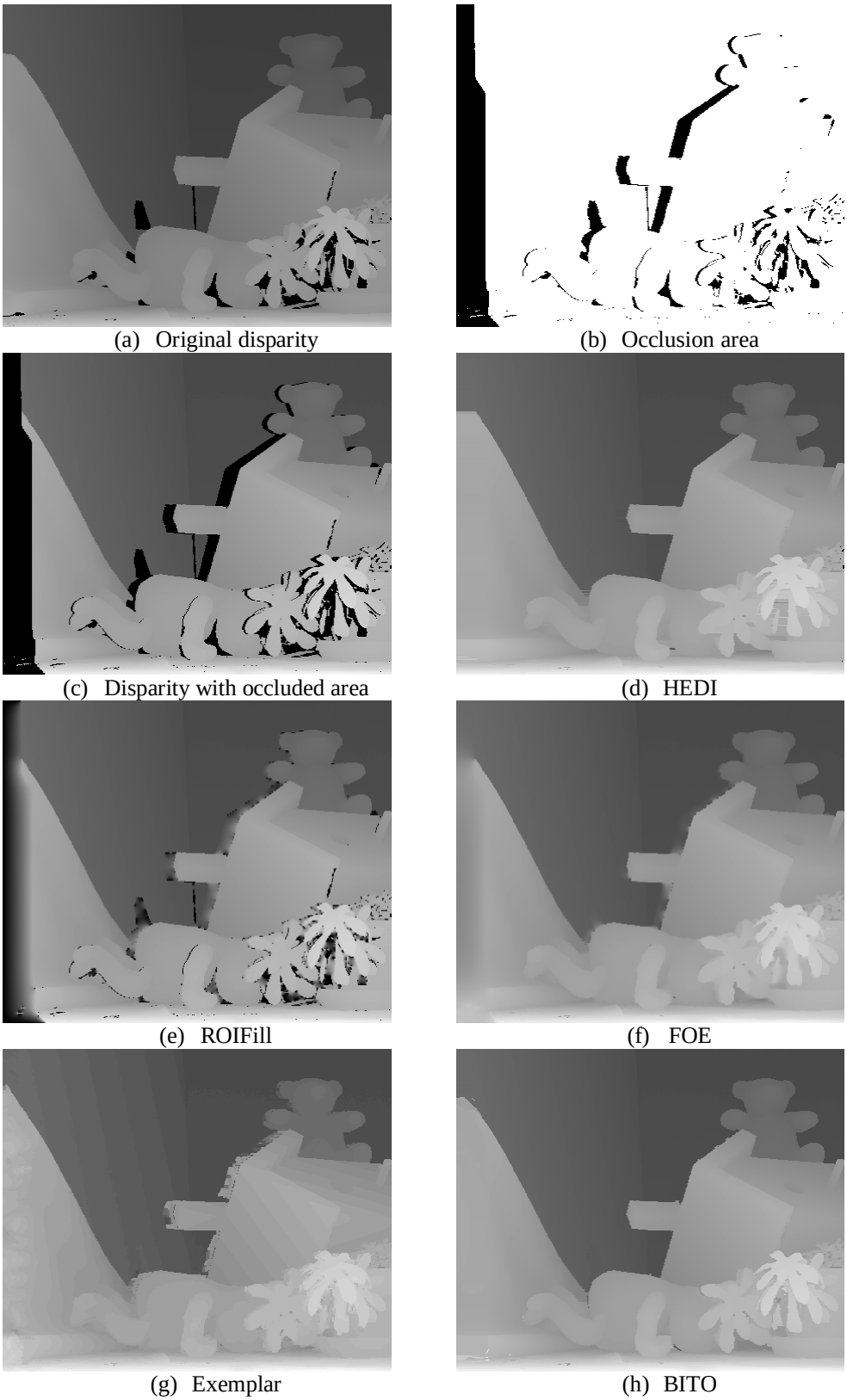
The performance of our proposed inpainting algorithm is tested to fill the occluded and unknown areas in the disparity image. The occluded area by its nature is large while the unknown area consists of small holes that cannot be calculated through disparity estimation. The size of the occlusion area plays an important role in the choosing of the *edge_threshold* parameter in addition to the amount of detail on that

area. If there are a lot of details (i.e. a lot of edges) or the occluded areas are large, a larger threshold is needed. This is because this threshold determines the filling-block size; a larger threshold consequently generates a larger block size that will speed up the inpainting process.

The proposed algorithm is compared to four well-known inpainting algorithms: HEDI, RIOFILL, Exemplar-based and FOE. Figure 5.10 shows the results of filling the occlusion area in the Teddy image using four previous inpainting algorithms (Figure 5.10 (d-g)) and they are compared to the proposed algorithm in Figure 5.10 (h). As we can see in Figure 5.10 (d), the occlusion areas are filled properly using the HEDI method even though their assumption is not applied in the case of filling the left- and right-most areas of disparity. Fortunately, such areas are filled from the foreground object, since there is no background, but without preserving edge discontinuities. The HEDI technique does not take into consideration the continuation of the edges, especially if there is a continuation for a curved object, as shown in Figure 5.11 (d) which is a close-up of Figure 5.10 (d).

According to the ROIFILL method, the occlusion area is considered too large for this technique. As we notice in Figure 5.10 (e), the occlusion area is not completely filled using the ROIFILL method; additionally, there is a high level of blurriness due to the smooth diffusion of the second derivative of the Laplacian equation. The FOE method is not specialized to inpaint large holes such as occlusion areas; thus, a high level of blurriness in the filled area in the far left of the disparity is generated, with a considerable level in the filled area in the gap behind the object. This blurriness is the cost of the intensity diffusion that does not take into consideration the texture.

As the Exemplar-based technique [94] is specialized to inpaint large areas, taking into consideration in the filling procedure the texture and the structure, the occluded area is completely filled, as shown in Figure 5.10 (g). Unfortunately, this technique is not specialized to fill the occluded area, so it fills the holes from both the background and the foreground object. Thus, the left-most filled area is inpainted properly since it is located from the side of the foreground object, while the filled gap behind the object is corrupted because it is a mixture of background and foreground.



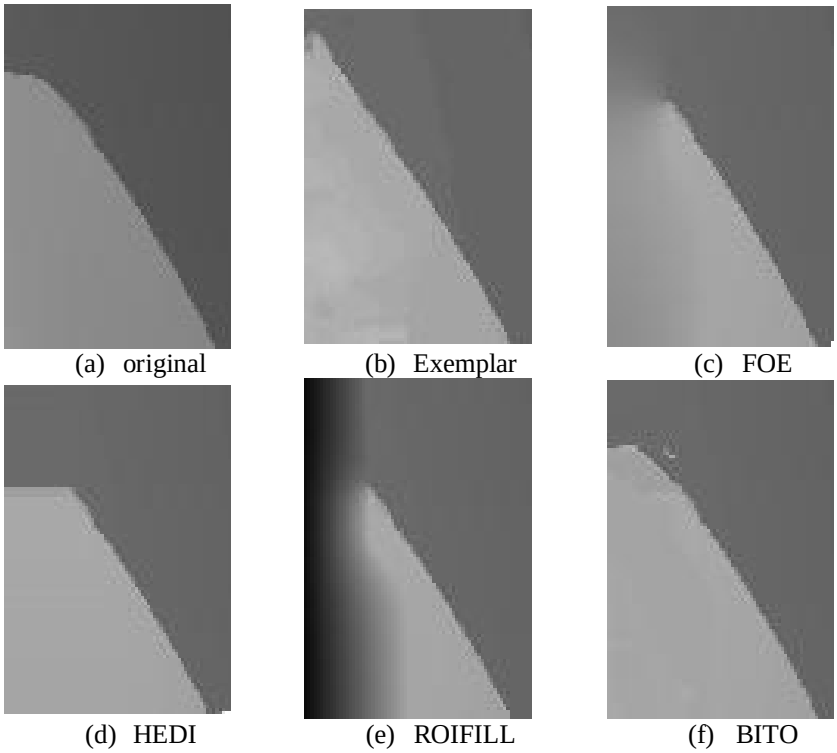
(a) Original disparity (b) Occlusion area

(c) Disparity with occluded area (d) HEDI

(e) ROIFill (f) FOE

(g) Exemplar (h) BITO

Figure 5.10: Shows the result of filling the occlusion area in the Teddy image by four previous inpainting algorithms (d-g) and they are compared to the proposed one in (h). (a) The ground truth disparity. (b) Results of the proposed occlusion detection algorithm. (c) The disparity map with occlusion areas.



(a) original (b) Exemplar (c) FOE

(d) HEDI (e) ROIFILL (f) BITO

Figure 5.11: Shows a close-up of the left-top of the inpainted disparity in Figure 5.10 (d-h) to clarify the accuracy of each technique.

We believe that the improved quality of the Exemplar-based technique [97] works better than Criminisi *et al's* technique since it is specialized for occlusion areas; however, it is still computationally expensive. The edges are continued very well but without taking care of the curved line, as noticed in Figure 5.11 (b), in contrast to our proposed method that preserves the line curvature, as shown in Figure 5.11 (f). Moreover, the proposed technique fills the two types of occluded area properly without any blurriness effect and preserving edge discontinuity. Figure 5.14 and Figure 5.16 and their close-ups in Figure 5.15 and Figure 5.17 are other examples of the occlusion inpainting in the Barn2 and Reindeer images. The Reindeer image is an example of a complicated occlusion area that shows the effectiveness of the proposed algorithm in recovering the occlusion area when compared to the other existing algorithms. The example presents a challenge to recover a hidden part of the curved line in the left side of the image that is difficult to reconstruct. As we note, HEDI cannot recover this area and it is filled from the background, while ROIFILL fails to inpaint most of the occluded areas. The Exemplar-inpainting technique fails to recover the correct shape of the curved line; moreover, it diffuses the intensities of

the false foreground object. The FOE technique inpaints these areas better than the previous algorithms but with a blurriness effect that mixes the foreground intensities with a background.

5.3.4.1 PSNR Results

For each different algorithm, the PSNR is calculated for the Teddy, Barn2, and Reindeer images shown in Table 5.1 and drawn in Figure 5.12 As we can see, the proposed BITO algorithm gained on average a higher PSNR of ~6, ~12, and ~17 db than the Exemplar, FOE and ROIFILL inpainting techniques, respectively. On the other hand, the average of PSNR results of the BITO technique is comparable to the HEDI, as BITO gained a higher PSNR of ~0.16 db than HEDI, which is considered a small difference. As we mentioned above, the unknown areas are excluded from the MSE calculation to produce a fair PSNR.

Table 5.1: Shows the PSNR for the Barn2, Reindeer and Teddy images under five different algorithms.

Technique name	Barn2	Teddy	Reindeer	Image size
BITO	34.8094	40.0404	32.6193	671 × 555
Exemplar	31.5956	31.3803	26.6126	447 × 370
FOE	28.6293	31.7672	10.2707	447 × 370
HEDI	35.8079	38.8843	32.2780	671 × 555
ROIFill	24.0081	18.5127	12.9065	671 × 555

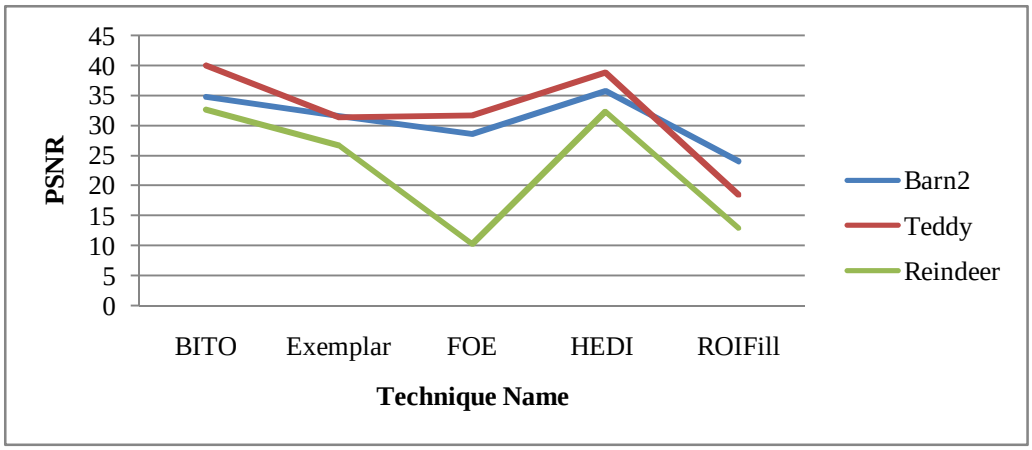


Figure 5.12: Shows the PSNR for Barn2, Reindeer and Teddy images under five different inpainting techniques.

5.3.4.2 Runtime comparison

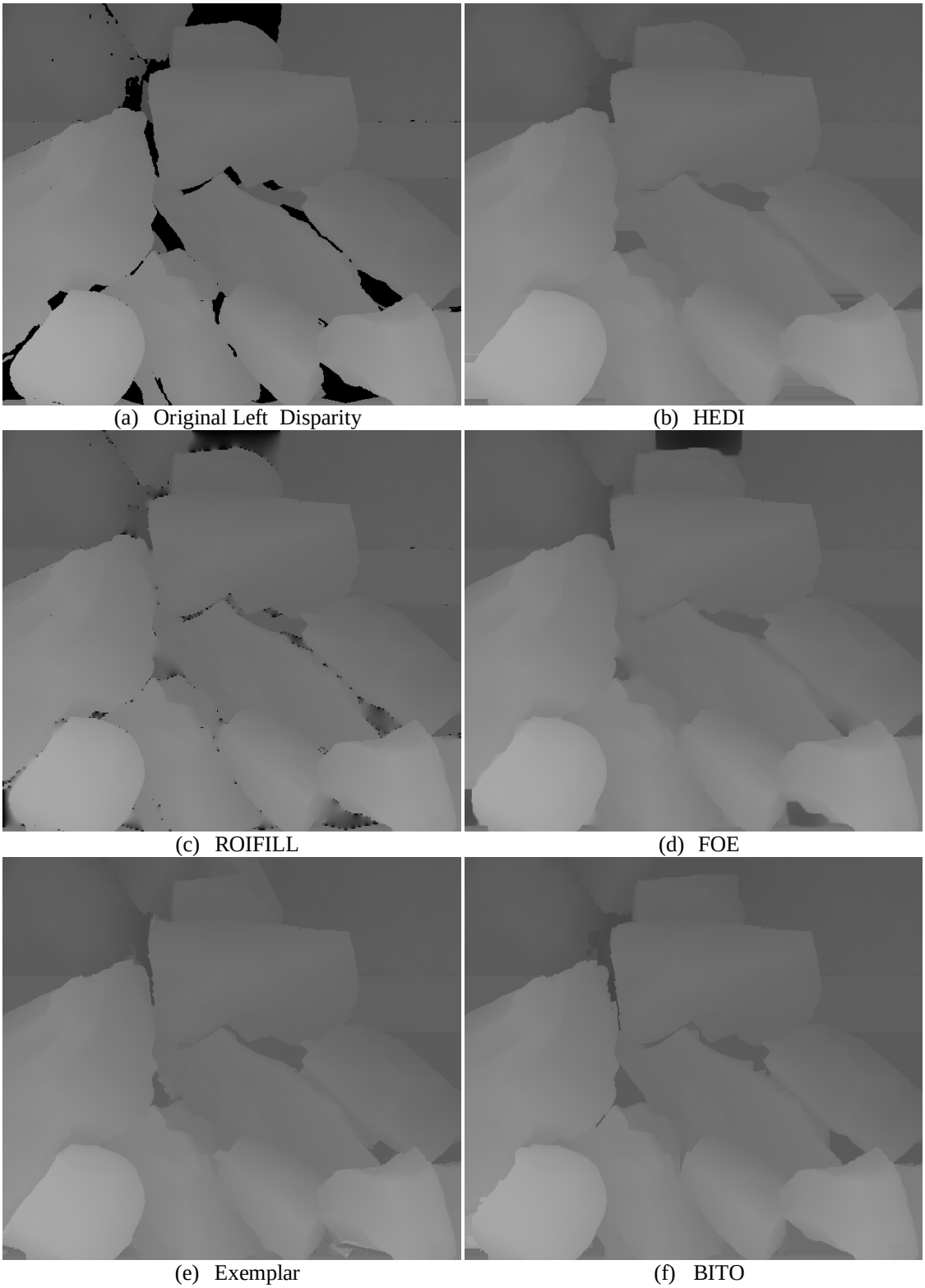
Inpainting runtime is considered an important issue in occlusion filling applications, such as multiview image interpolation for multiscopic 3D display. In Table 5.2, the runtime is calculated for five inpainting techniques that inpaint the occlusion areas in the Teddy image. The HEDI and ROIFILL methods are considered to have the lowest time complexity since they depend on a simple assumption, while the Exemplar-based and FOE techniques are time-expensive methods. Inpainting using the proposed technique takes a considerable time when compared to the Exemplar-based and FOE techniques. We used a 450 × 375 image size for the Teddy dataset because, were we to use a larger image, the occlusion area would be extended and we could not calculate the time duration for the Exemplar-based and FOE techniques for out-of-memory reasons. Table 5.3 depicts another runtime comparison for five inpainting techniques that inpaint the occlusion areas in the Reindeer image.

Table 5.2: Shows the runtime required to fill the occlusion areas in the Teddy image of size 450 × 375 for each technique in seconds using a laptop Intel Core 2 Duo CPU @ 2.00GHz 2.00GHz, and 3.00 GB RAM.

Technique name	Required Time
HEDI	0.236050
ROIFill	0.206879
Exemplar	433.114688
FOE	436.692172
BITO	5.196117

Table 5.3: Shows the runtime required to fill the occlusion areas in the Reindeer image for each technique in seconds using a laptop Intel Core 2 Duo CPU @ 2.00GHz 2.00GHz, and 3.00 GB RAM.

Technique name	Required Time	Image size
HEDI	0.413368	671 × 555
ROIFill	1.380815	671 × 555
Exemplar	304.611208	447 × 370
FOE	434.515383	447 × 370
BITO	19.448148	671 × 555



(a) Original Left Disparity (b) HEDI

(c) ROIFILL (d) FOE

(e) Exemplar (f) BITO

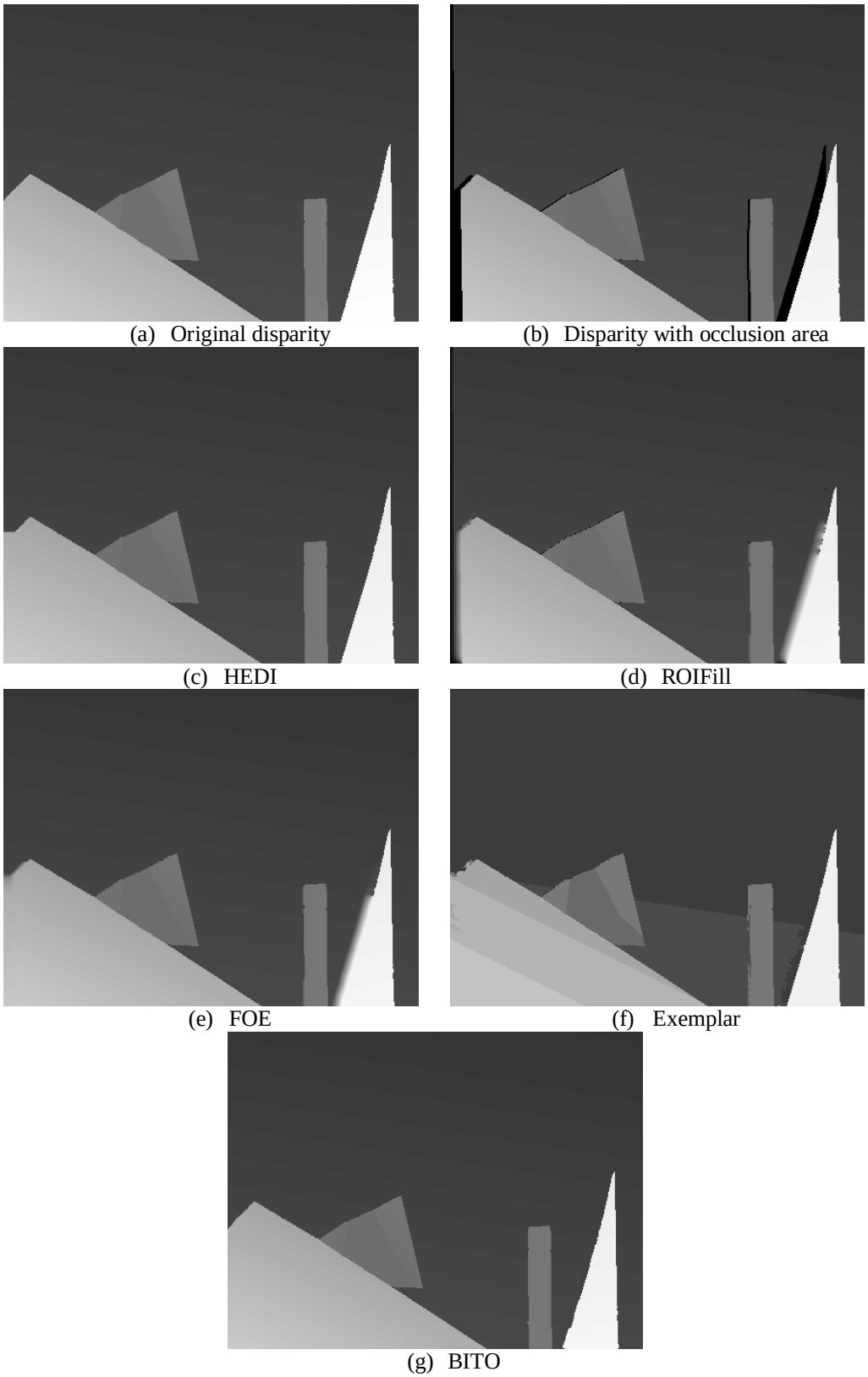
Figure 5.13: Shows the result of filling the unknown areas in the Rocks image by four previous inpainting algorithms (b-e) and they are compared to the proposed one in (f). (a) The ground truth disparity that comes with unknown areas.

5.3.4.3 Inpainting unknown areas

Inpainting an unknown area is considered a simpler issue than inpainting an occluded area since it is smaller. In general, when we compare the result of our inpainting technique with the other methods, as in Figure 5.13, we can see that the proposed algorithm inpaints the unknown area accurately. However, other techniques work much better in filling the unknown area due to the small size of the holes when compared to filling the occlusion areas. PSNR metric cannot be calculated for inpainting unknown areas due to unavailability of a ground truth of such areas. In Table 5.4, the runtime is calculated for five inpainting techniques that inpaint the unknown areas in the Rocks image.

Table 5.4: Shows the runtime required to fill the unknown areas for each technique in seconds using a laptop Intel Core 2 Duo CPU @ 2.00GHz 2.00GHz, and 3.00 GB RAM and image size of 638 × 555.

Technique name	Required Time
HEDI	0.191961
ROIFill	0.130977
Exemplar	474.460757
FOE	913.400221
BITO	1.712



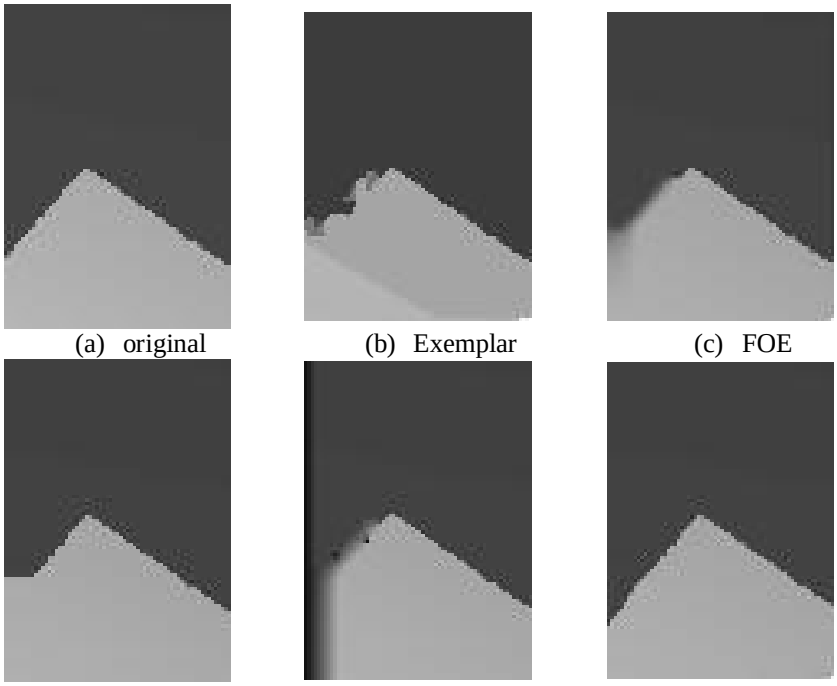
(a) Original disparity (b) Disparity with occlusion area

(c) HEDI (d) ROIFill

(e) FOE (f) Exemplar

(g) BITO

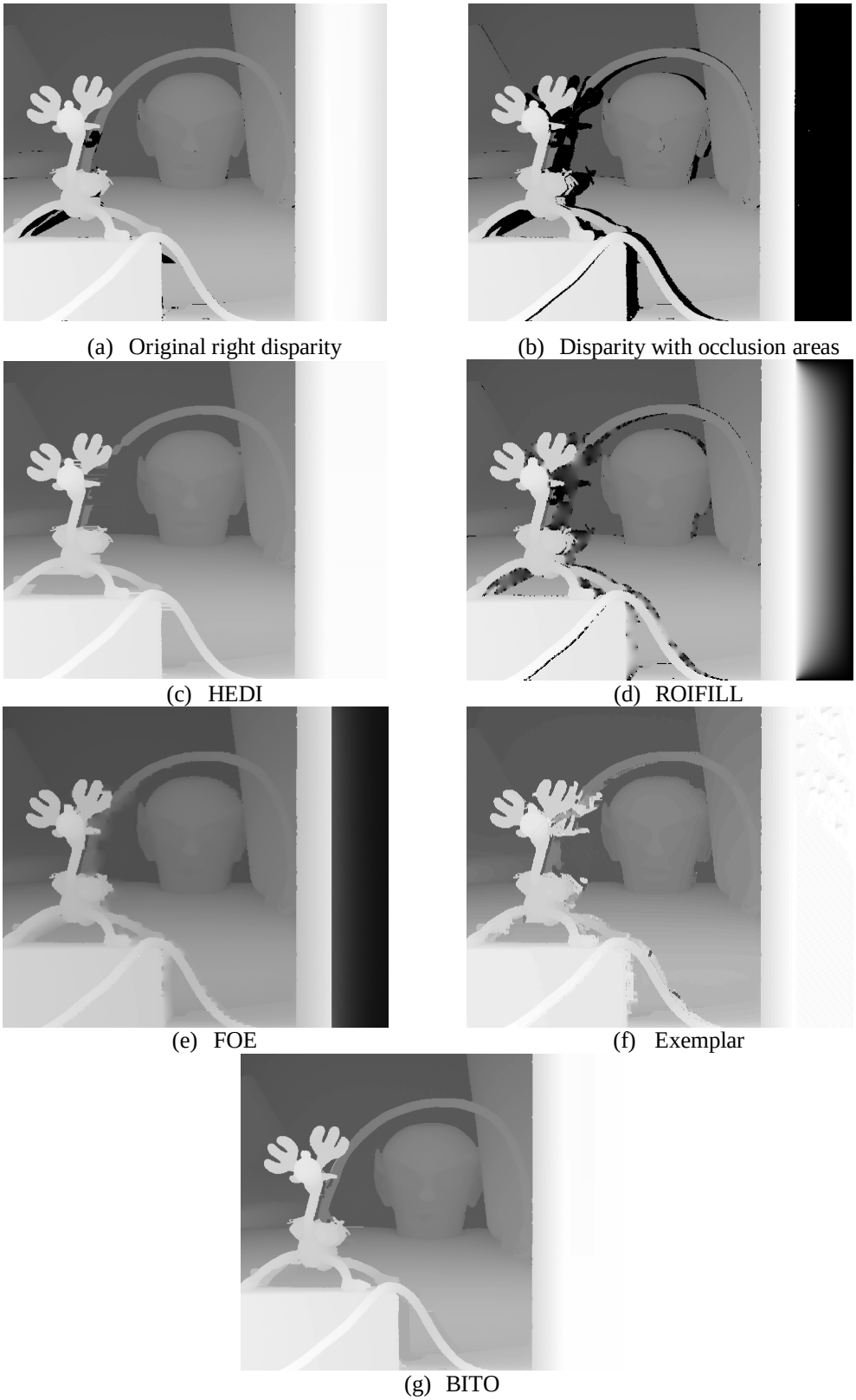
Figure 5.14: Shows the result of filling the occlusion area in the Barn2 image by four previous inpainting algorithms (c-f); they are compared to the proposed one in (g). (a) The ground truth disparity. (b) The disparity map with occlusion areas.



(a) original (b) Exemplar (c) FOE

(d) HEDI (e) ROIFILL (f) BITO

Figure 5.15: Shows the close-up of the left-most area of the inpainted disparity in Figure 5.12 (c-g) to clarify the accuracy of each technique.



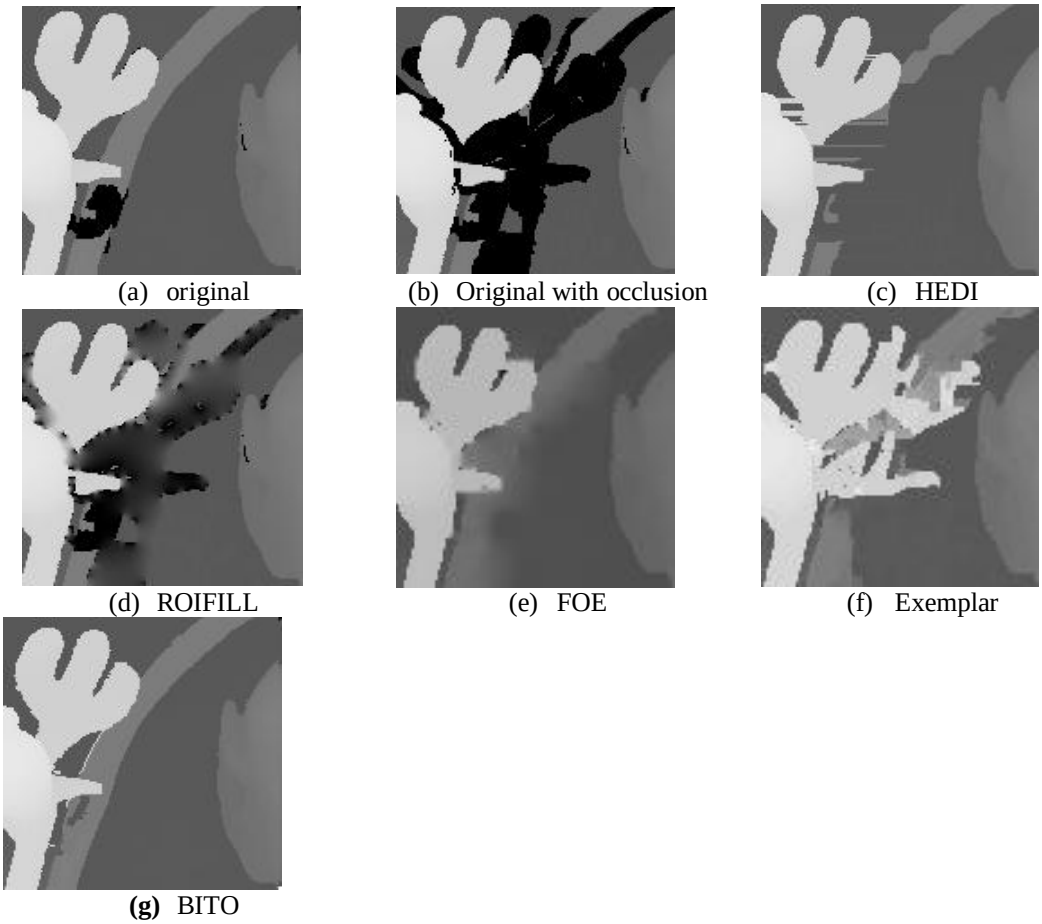
(a) Original right disparity (b) Disparity with occlusion areas

(c) HEDI (d) ROIFILL

(e) FOE (f) Exemplar

(g) BITO

Figure 5.16: Shows the result of filling the occlusion area in the Reindeer image by four previous inpainting algorithms (c-f); they are compared to the proposed one in (g). (a) The ground truth disparity. (b) The disparity map with occlusion areas.



(a) original (b) Original with occlusion (c) HEDI

(d) ROIFILL (e) FOE (f) Exemplar

(g) BITO

Figure 5.17: Shows the close-up of a complicated area in the inpainted disparity in Figure 5.14 (c-g) to clarify the accuracy of each technique.

5.4 Summary

In this chapter, a novel inpainting technique has been proposed to overcome the deficiencies of previous inpainting techniques. This novel technique is specialized for occlusion area filling in disparity images. We have compared our algorithm results with four well-known inpainting algorithms qualitatively and quantitatively using PSNR; the proposed algorithm results outperform these algorithms. One of them is specialized for occlusion filling and the other three are for general image inpainting, specifically for object removal or image restoration. Our algorithm has many advantages; For example, the occluded area is filled without any blurriness since it does not depend on any smoothness diffusion. By exploiting the edge map of the underlying stereo image, edge discontinuities are diffused in the occlusion area in the disparity map, and the line curvature is continued. We performed various

analyses and computer simulations which confirmed that the proposed technique inpaints the occlusion area with high accuracy. This is shown using PSNR calculation, where the proposed BITO algorithm gained on average a higher PSNR of ~0.16, ~6, ~12, and ~17 db than Exemplar, HEDI, FOE, and ROIFILL inpainting techniques, respectively.

CHAPTER

6

Occlusion-Aware Intermediate View Reconstruction

In Chapter 4, we proposed a novel occlusion detection algorithm, and in Chapter 5 we proposed another novel algorithm to Inpaint those areas. In this chapter, we will employ these proposed algorithms in a novel occlusion-aware intermediate view reconstruction. This novel algorithm focuses on how to find an intermediate image with a given disparity between two input images.

An improvement in the backward-projection of the intermediate view is proposed by adding occlusion awareness to the reconstruction algorithm and applying three quality improvement techniques to reduce image artifices. First, we remove holes in the virtual disparity produced by image re-sampling. Second, we remove ghost contours from the virtual view and restore them to their correct place. Third, since the novel reconstruction algorithm detects and inpaints the occlusion areas in the two input disparity maps, the disocclusion areas are recovered from one of the input images.

We compare the proposed reconstruction algorithm to the standard backward- and forward-projection approaches quantitatively using PSNR, and qualitatively by conducting a human-trial experiment on a 3D display.

6.1 Introduction

Intermediate view reconstruction is a widely-used technique to generate intermediate images from a few existing views and their corresponding disparities. This is accomplished as if we have positioned virtual cameras between a few real cameras to create virtual views at those positions. The disparity map is used to displace image pixels from the existing views to a new location in a virtual view. Unfortunately, when the object occupies a new place, it will leave behind a gap called a disocclusion area or newly-exposed area.

The generation of the intermediate image from stereoscopic images is not straightforward, due to the ambiguity of the occluded region and the need to preserve the discontinuity of the edges. Therefore, many studies have been conducted in attempts to generate the intermediate images with high quality. These studies can be classified into two main categories based on how the disparity that compensated into the intermediate images is calculated: backwards-disparity compensation [73, 75] and forward-disparity compensation [76].

In backward-disparity compensation, the disparity values are defined on the sampling grid of the intermediate image (unknown image) in the direction of both left and right images (known images). These backward-projection algorithms use different ways to reconstruct the intermediate view points from different known images; some of them use linear interpolation [135, 136], which produces a blurred image, while one of them uses non-linear interpolation [137], which generates a patchy effect on the intermediate image.

In contrast, the structure of the intermediate image using the forward disparity compensation approach is defined on the sampling grid of one or all of the available images. A problem occurs when the available images are pointing to the irregular samples of the intermediate image. In this case, simple methods are used to solve the problem, such as rounding, which is not effective [76]. Recently, a spline-based reconstruction has been proposed; this is an effective model for this problem but it is computationally expensive [71].

In this chapter, an improvement to the backward-projection of the intermediate view is proposed by adding occlusion awareness to the reconstruction algorithm. In addition, we apply three quality improvement techniques to reduce image artifices generated through formation of the intermediate view. The reasons for generating these image artifices and the state-of-the-art solutions to reduce them are discussed in the next section.

6.2 The Problematic Areas of Reconstruction Process and Their Existing Solution

A number of image artifices are generated through formation of the intermediate view and they must be removed; they include the generation of holes and cracks due to image resampling, generation of ghost contours due to disparity's sharp edges, and the appearance of disoccluded areas due to cameras changing position [72].

6.2.1 Image Holes and Cracks through Re-sampling Process

Building a disparity map for the intermediate image from one of the known images will result in small holes and cracks in the disparity map. It is necessary to fill these holes to generate an error-free intermediate image. These holes are generated from rounding image coordinates to the nearest position in the sampling grid, whereas the sampling grid for any image is defined in positive integers; the majority of the transformed disparity coordinates are not integers and require rounding to the nearest integer, leaving an empty pixel. For example, as pixel 24 is filled, and the following coordinate is 25.7, then rounding this coordinate to the nearest coordinate in the sampling grid will result in filling pixel 26 and leaving 25 empty. These cracks and holes are clearly shown in Figure 6.8 (c).

A majority of researchers fill these holes by applying a median filter to the disparity area [72, 111, 113, 115]. A median filter does not work well when filling a hole in a detailed area. Moreover, by using a median filter (3 × 3) which is the suitable size to fill the holes in the disparity map [72], a hole of size larger than 2 × 2 will not be filled and will be left empty.

6.2.2 Ghost Contours Due to Disparity's Sharp Edges

Generally, since texture images have a high resolution, an object's edges cover two to three pixels in the image while, in the disparity map, an object's edges are very sharp and are covered by one pixel resolution. Consequently, pixels from foreground objects will interpolate falsely in the background, appearing as a ghost in the virtual image, as shown in Figure 6.5 (c) and its close-up in (d). Figure 6.1 demonstrates how these contours appear in the interpolated image.

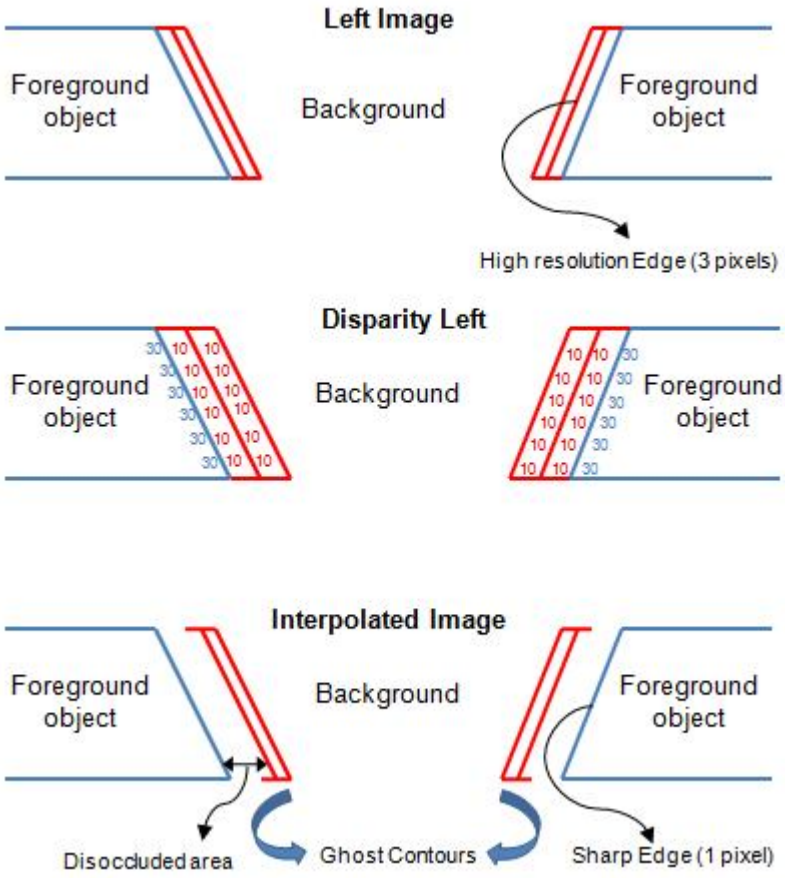


Figure 6.1: Demonstrating ghost contours problem

Some researchers have labelled these edges in the texture image to avoid interpolating these pixels to a virtual image [72]; this solution successfully removes ghost contours but leaves the interpolated edges sharp with low resolution. Another solution is to use disocclusion dilation [113], which extends the disocclusion region one pixel; this is not accurate because most of the ghost contours' widths are two to

three pixels, leading to ghost contours being left in the interpolated image. Moreover, dilation from both sides of the disocclusion region will corrupt the image since the ghost contour occurs from one side of the object; this leads to the interpolation of incorrect pixels in a virtual image from the reference image.

6.2.3 Disocclusion Area Due to Camera Changing Position

Due to changing camera position, image objects will shift a number of pixels depending on how close they are to the camera; the closer they are, the larger the shift. This pixel shifting leaves an uncovered area in the virtual image called disocclusion or newly-exposed area. Recovering those areas is considered an important challenge in reconstructing the intermediate image. The size of disocclusion areas depends on the camera baseline; filling a larger region is considered a harder task.

Many filling techniques have been proposed in the literature [72, 102, 103], and a comprehensive comparison for disocclusion filling techniques has been conducted [103] to show which one is the best for those areas. The comparisons confirm that using an inpainting technique to fill those areas is very expensive, as it sometimes takes a few hours and is not accurate enough, especially as it is not designed to fill the disparity map. In particular, filling disocclusion regions with horizontal interpolation using depth map technique provides the best intermediate image quality. This technique fills the disocclusion area only from the background by assuming that the area belongs to the background. To summarize, this technique works well only when the foreground object is superimposed on a background, while it fails when another object and part of the background lie underneath the foreground object. Therefore, continuing the edges of the underlying object is necessary to undistort the virtual image. Moreover, it fails when a disocclusion area is located in the far left/right side of the image and it is necessary to recover those parts of the object and background.

6.3 The Proposed Intermediate View Reconstruction Algorithm

The backward-projection approach is considered advantageous when compared to the forward-projection approach as the backward-projection approach assigns an

intensity value for each image point in the intermediate image, and the resulting image does not have any black cracks on it. In the forward-projection approach, some image points are assigned to the same location in the virtual view and the other points do not have intensity values, producing black cracks in the image. Moreover, a study [71] confirmed that the backward-projection approach is considered more reliable, if there are errors in the disparity estimation, than the forward-projection approach. Therefore, we propose a novel reconstruction algorithm that exploits the advantages of backward projection and reduces its drawbacks by adding occlusion awareness and reducing image artifices.

The framework of the proposed algorithm is depicted in Figure 6.2, illustrating the pipeline of the interpolation algorithm; the framework consists of input images, interpolation steps, output image, and post-processing step. The input to the proposed algorithm consists of four images: two reference images and their disparities. The reference images are a stereoscopic pair (left image I_L and right image I_R) captured by real cameras. In regard to the disparity maps, we used a given disparity downloaded from different datasets, since this study focuses on a high-performance reconstruction process. The output from the interpolation steps will be the intermediate image at a certain position between the reference images (I_L and I_R).

In the following section we will explain how the proposed reconstruction algorithm generates an intermediate image of high quality step by step:

***Step #1*: Detect the occlusion area**

We use our proposed occlusion/newly-exposed detection algorithm, which was explained in detail in Chapter 4 since it detects the occluded areas with high accuracy and fewer errors. Moreover, it is consistent under different types of images

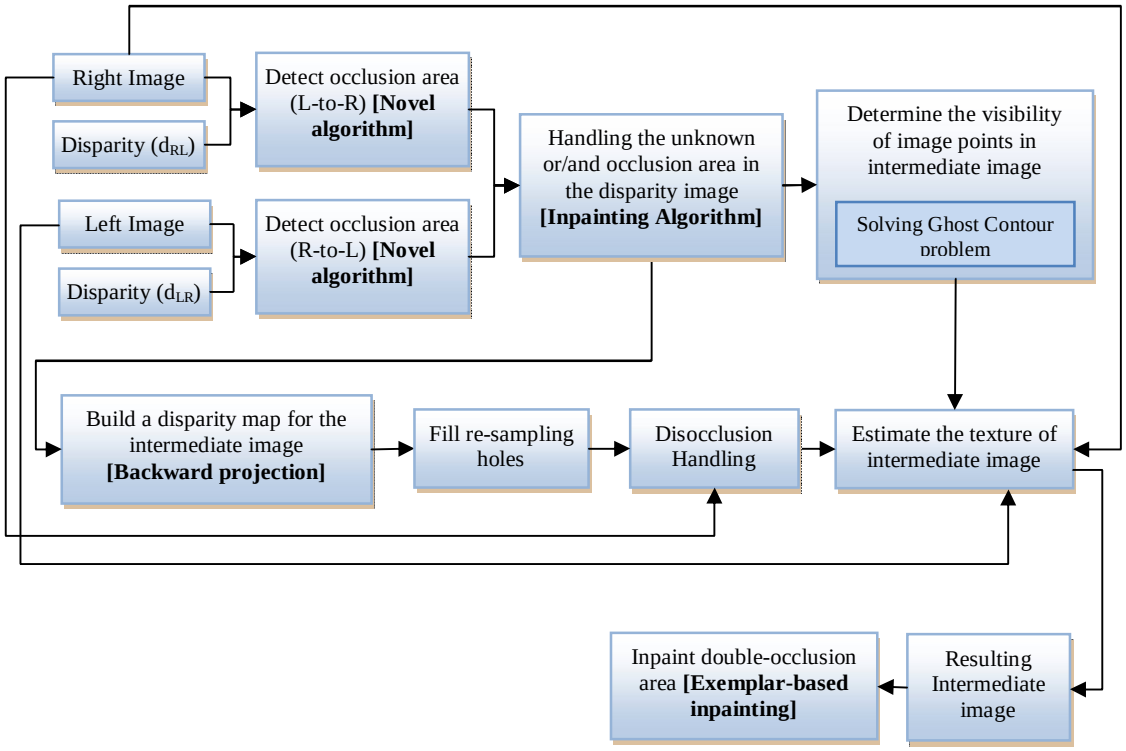


Figure 6.2: Showing the general framework for the proposed intermediate view reconstruction algorithm

and matching algorithms due to a high diversity of input images to the interpolation algorithm. We detect the occlusion areas in two directions from left-to-right and from right-to-left to inpaint those in the following step. This is illustrated in the following equations:

$$NE_{L-to-R} = OD_{left}(d_L), \tag{6.1}$$

$$NE_{R-to-L} = OD_{Right}(d_R), \tag{5.7}$$

where d_L and d_R are the disparity from left-to-right and from right-to-left, respectively. OD_{Left} and OD_{Right} refer to the occlusion detection algorithm which computes the newly-exposed areas $NE_{L\text{-}to\text{-}R}$ and $NE_{R\text{-}to\text{-}L}$, respectively.

***Step #2*: Inpaint the occlusion and unknown area in the disparity map**

The disparity estimation methods force a disparity to be calculated even in the occlusion area by choosing the best estimate. Therefore, if the disparity is estimated,

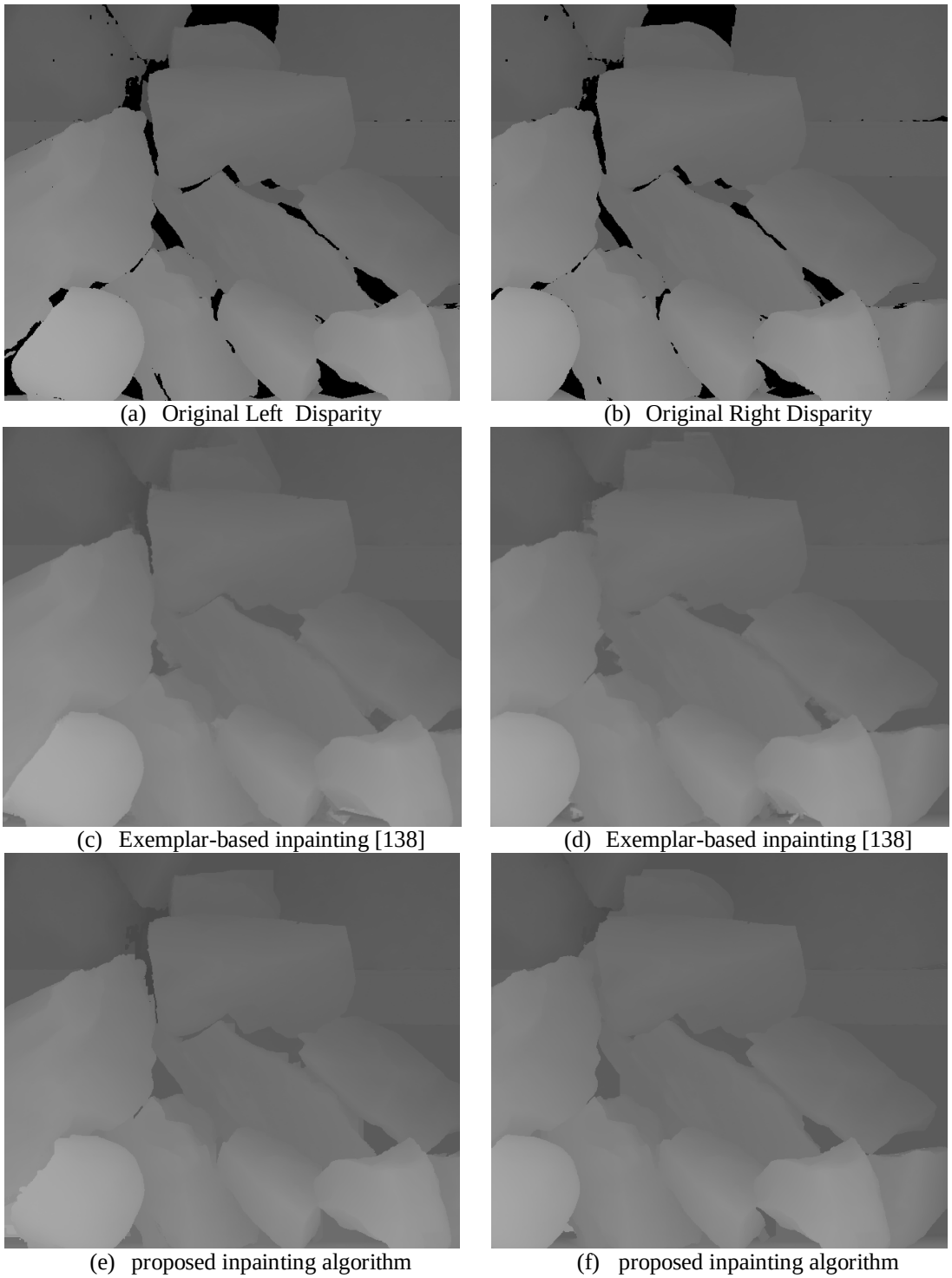
it is necessary to re-inpaint those areas while, if it is computed by a computer program, the occluded areas are filled from the other disparity. In addition, this kind of disparity leaves an empty area without disparity, which is called an unknown area. For reconstruction purposes, filling those areas is necessary to interpolate an accurate intermediate image. The occluded and unknown areas are filled with our proposed disparity inpainting algorithm, explained in Chapter 5 since it recovers the empty area by preserving the object edges inherited from the underling stereo image.

$$Id_{LR} = Inpaint_{d_L}(I_L, d_L, NE_{R\text{-}to\text{-}L}) \quad (6.2)$$

$$Id_{RL} = Inpaint_{d_R}(I_R, d_R, NE_{L\text{-}to\text{-}R}) \quad (6.3)$$

Id_{LR} and Id_{RL} are the inpainted left and right disparities, respectively. $Inpaint_{d_{LR}}$ and $Inpaint_{d_{RL}}$ refer to the inpainting operations applied to the d_L and d_R, respectively. $NE_{L\text{-}to\text{-}R}$ and $NE_{R\text{-}to\text{-}L}$ are the newly-exposed areas resulting from the previous step. These areas are added to the disparity map as zero intensity to refill them with a more accurate disparity map. The input for the inpainting algorithm to inpaint the unknown area will be without the newly-exposed area: $Inpaint_{d_L}(I_L, d_L)$ and $Inpaint_{d_R}(I_R, d_R)$. This is because the unknown areas are zero pixels in the original disparity map that need inpainting in different places, as shown Figure 6.3. Specifically, the black pixels that appear in the left and right disparities in Figure 6.3 (a) and (b), respectively, are the unknown areas which need inpainting.

Inpainting those areas using the Exemplar-based algorithm and the proposed algorithm is demonstrated in the second and third rows of Figure 6.3, respectively. As we can see in this figure, the proposed algorithm fills those areas more accurately than the Exemplar-based algorithm even though the inpainting time of the latter algorithm is 474.5 seconds, while the proposed one takes 1.7 seconds to inpaint those areas. Thus, the proposed inpainting algorithm is more accurate and faster than the Exemplar-based one. A comparison with three other techniques of inpainting the unknown and occlusion areas was conducted in the previous chapter.



(a) Original Left Disparity (b) Original Right Disparity

(c) Exemplar-based inpainting [138] (d) Exemplar-based inpainting [138]

(e) proposed inpainting algorithm (f) proposed inpainting algorithm

Figure 6.3: Showing results of inpainting unknown areas by the Exemplar-based algorithm in the second row and the proposed algorithm in the third row.

Step #3: Determine the visibility of image points in intermediate image

Capturing different images from different positions leads to the visibility of some regions in one view and their disappearance in the other view. Consequently, the intermediate image will have some regions that are visible in just one of the surrounding images, as in Figure 6.4. Therefore, it is necessary to determine the visibility of image points of the intermediate image to reconstruct the correct information from the correct source. Fortunately, we can use the proposed occlusion algorithm for this task. Since the full occlusion area is detected between I_L and I_R using their disparities, we can find the partial occlusion area by multiplying the disparity maps by α and (1 – α), if the intermediate image is located between I_L and I_R at position α. This is illustrated in the following equations:

$$NE_{L-to-Int} = OD_{left}(\alpha * d_L) \tag{6.4}$$

$$NE_{Int-to-R} = OD_{Right}((1-\alpha) * d_R) \tag{6.5}$$

Practically, we will give an example to illustrate how the proposed occlusion algorithm detects the large and partial occlusion areas between the images, as shown in Figure 6.4. Using d_L as input to the equation (6.4) yields areas C and D as newly-exposed areas in I_R, while using d_L multiplied by α as input gives us area D as a newly-exposed area from left-to-intermediate; this means area D is a newly-exposed area in the intermediate image and is not visible in I_L. Consequently, we have to reconstruct it from I_R. Correspondingly, using d_R as input to equation (6.5) results in areas A and B as newly-exposed in I_L, while a partial newly-exposed area in I_{Int} will be A; in this area A is not visible in I_R and we should use I_L to recover this area.

Finally, we will construct a visibility matrix VM that contains the visibility label for each image point in the intermediate image, whether they are visible in I_L, I_R, both or neither, and label them as 1, -1, 0 and 2, respectively. The values of the VM matrix are created based on $NE_{L\text{-to-Int}}$ and $NE_{Int\text{-to-R}}$ matrices as shown in equation (6.6), where the sampling grid of VM is defined on the same sampling grid of the NE matrices.

$$VM(x) = \begin{cases} 2 & , \ if \quad NE_{L-to-Int}(x) = 1 \ and \ \ NE_{Int-to-R}(x) = 1 \\ 1 & , \ if \quad NE_{L-to-Int}(x) = 1 \\ -1 & , \ if \quad NE_{R-to-Int}(x) = 1 \\ 0 & , \ if \quad otherwise \end{cases} \tag{6.6}$$

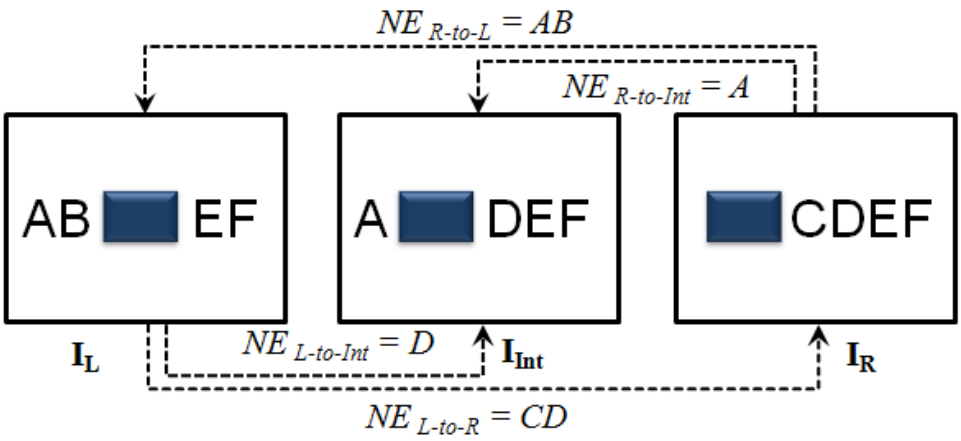
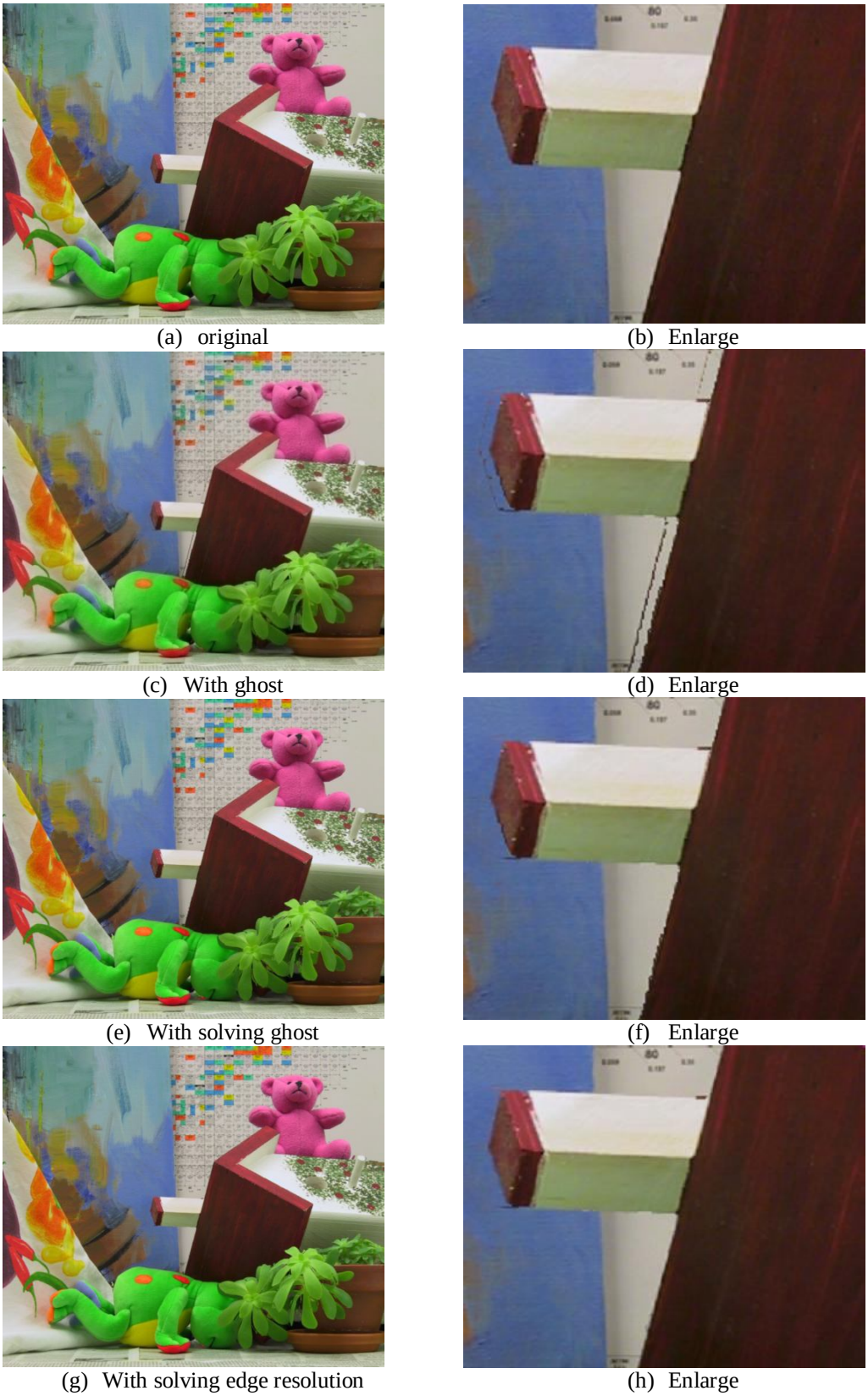


Figure 6.4: Showing newly-exposed area principle

***Step #4*: Removing ghost contours**

In regard to the ghost contour artifices that appear in the intermediate image in Figure 6.5 (c) and its close-up in (d), we sort them out simultaneously with a calculation of the intermediate image visibility at an early stage. As the visibility matrix labels the source of each image point in the intermediate image, pixels that cause ghost contours are labelled by adding three pixels to the newly-exposed area that will be covered from the surrounded images. This extension will be from one side of the newly-exposed area in the direction where the ghost contour is located. Specifically, if the ghost contour is located to the right of the foreground object, the newly-exposed area will extended by three pixels from the right while, if it is located to the left of the foreground object, the newly-exposed area will extended from the left three pixels.

Because the foreground object edges cover two to three pixels in the texture images, extending the newly-exposed area by one pixel leaves some ghost contours in the interpolated image. Therefore, we extend the newly-exposed area by three pixels since this extension is from the side of the background and it is visible in both



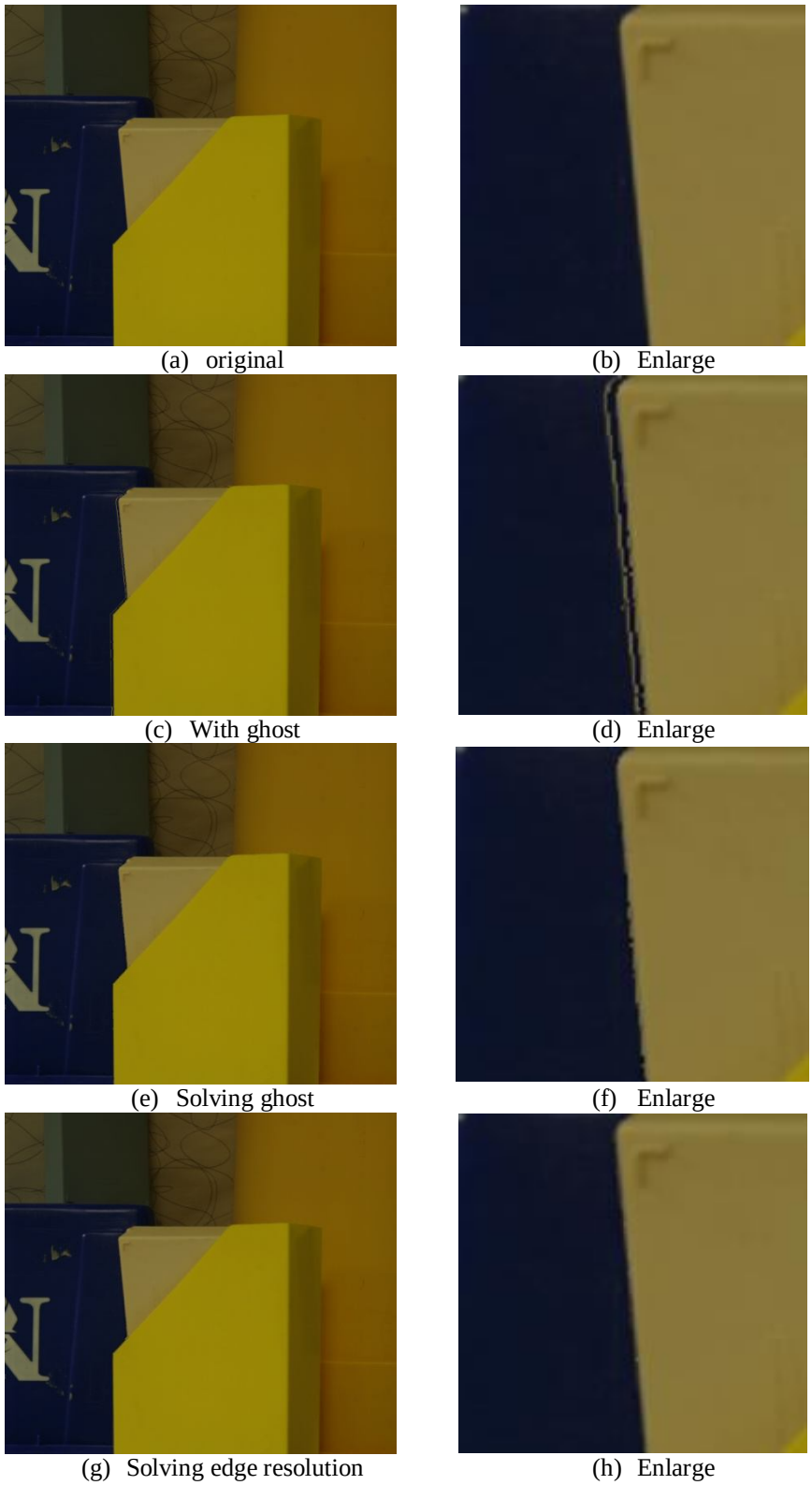
(a) original (b) Enlarge

(c) With ghost (d) Enlarge

(e) With solving ghost (f) Enlarge

(g) With solving edge resolution (h) Enlarge

Figure 6.5: Showing results of the proposed ghost removal technique



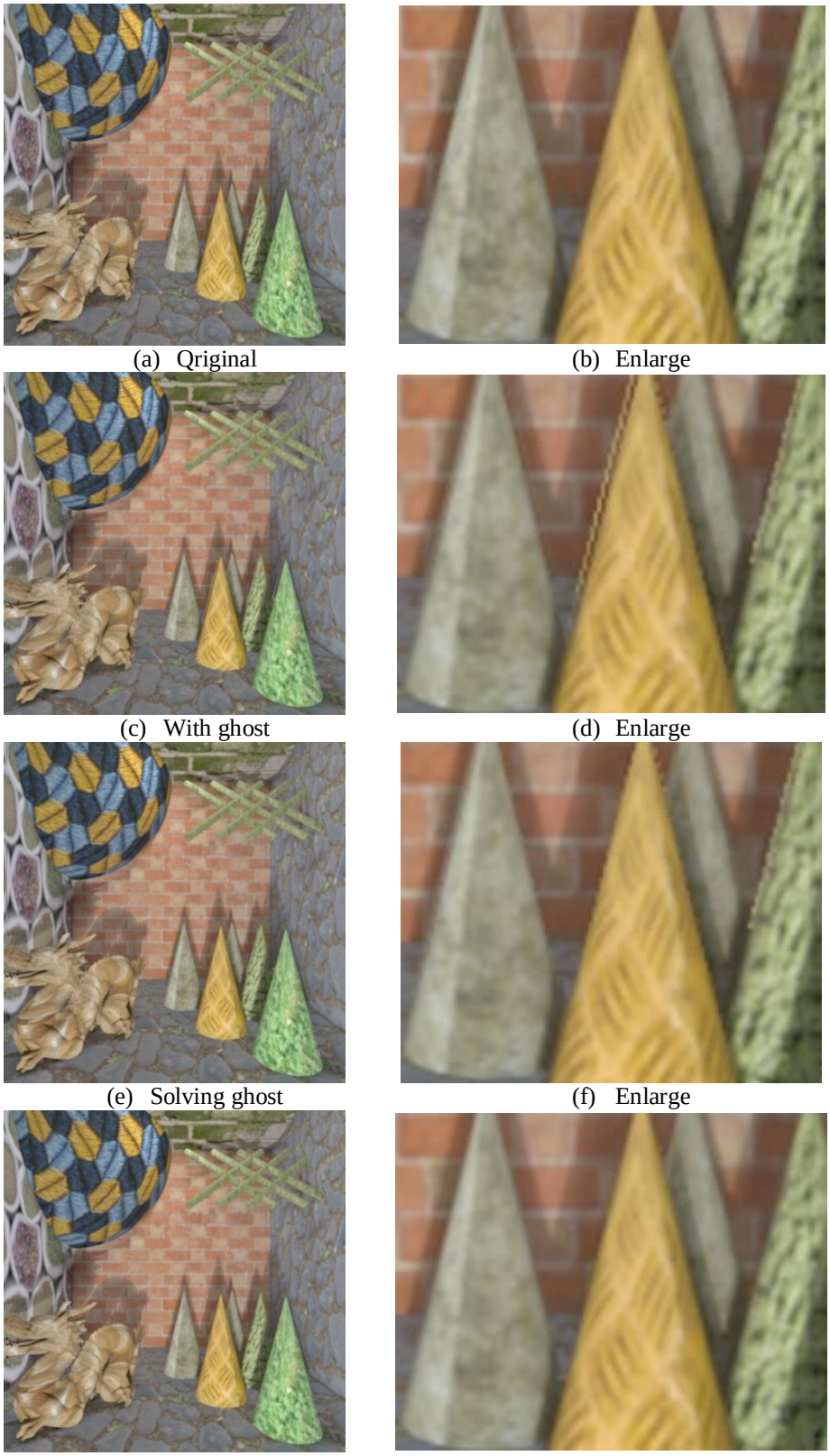
(a) original (b) Enlarge

(c) With ghost (d) Enlarge

(e) Solving ghost (f) Enlarge

(g) Solving edge resolution (h) Enlarge

Figure 6.6: Showing results of the proposed ghost removal technique



(a) Qriginal (b) Enlarge

(c) With ghost (d) Enlarge

(e) Solving ghost (f) Enlarge

(g) Solving edge resolution (h) Enlarge

Figure 6.7: Showing results of the proposed ghost removal technique

images; consequently all the ghost contours disappear from the interpolated image without leaving any corruption. Noticeably, omitting these edges from the interpolated image will generate an annoying cut-off for the foreground objects, as shown in Figure 6.5 (f), thus yielding a low resolution image. Therefore, it is necessary to recover the edge resolution in the interpolated image; this will be addressed at a later stage.

Figure 6.5 and Figure 6.6 are examples of ghost contours on real images while Figure 6.7 is another example on a synthetic image. These figures compare the original virtual image edges in (a) and its close-up in (b) with the reconstructed virtual image edges after removing ghost contours and enhancing edge resolution in (e) and (g) and its close-up in (f) and (h), respectively. Enhancing the edge resolution process is illustrated in the subsequent steps.

***Step #5*: Build a disparity map for the intermediate image**

Simply, we can create an intermediate image disparity map (d_{Int}) based on the surrounding reference disparity maps d_L and d_R. This is done by transferring the coordinates of d_L or d_R into d_{Int} coordinates. Specifically, in our proposed reconstruction algorithm, we use d_L to build d_{Int} by defining d_L, d_R, and d_{Int} on the sampling grid S_L, S_R, and S_{Int}, respectively; then the d_{Int} is calculated as in equation (6.7) and (6.8):

$$d_{Int}(x - \alpha d_L(x)) = d_L(x) \qquad \forall x \in S_L \tag{6.7}$$

$$d_{Int}(x + (1-\alpha) d_R(x)) = d_R(x) \qquad \forall x \in S_R \tag{6.8}$$

Unfortunately, $x - \alpha\, d_L(x)$ and $x + (1 - \alpha)\, d_R(x)$ may not belong to the sampling grid S_L or S_R, respectively. Therefore, we use a round operation to let $x - \alpha\, d_L(x)$ and $x + (1 - \alpha)\, d_R(x)$ belong to S_L or S_R, respectively. Using a round operation leaves small black holes and cracks in the d_{Int} in addition to the holes in the disocclusion area generated by changing virtual camera position; this will handled in the subsequent stages. These holes and cracks are clearly shown in Figure 6.8 (c).

***Step #6*: Filling re-sampling holes**

Accurate filling of the holes and cracks produced by the transfer of disparity maps from a known to a virtual one is required in order to generate an error-free intermediate image. As we described in section 6.2, a median filter is one of the solutions applied in the literature to fill these holes and cracks, but we proved in this study that a median filter gives inaccurate results in detailed areas of the image. Moreover, the median filter leaves holes of size greater than 2 × 2 pixels unfilled. Therefore, we propose a new filling technique that overcomes the median filter disadvantages.

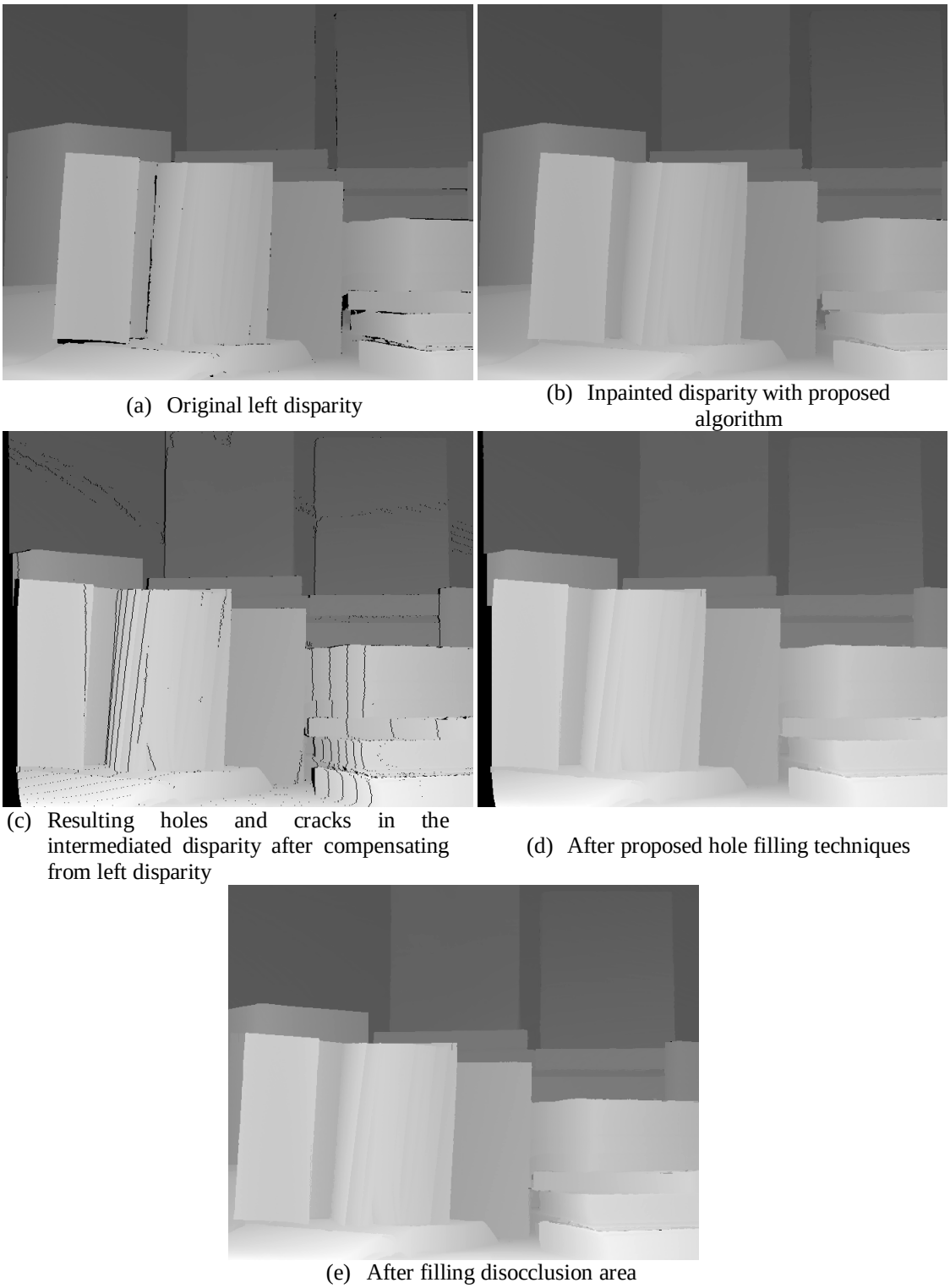
Our proposed technique is an object-based filling that considers the holes located in a foreground object or around its edges as belonging to the object and filled by its intensity while, if it is located in the background, it will be filled by the background intensity. Having omitted, in the previous step, two to three pixels from the object's edges in the interpolated image, considering any hole located around an object's edges as belonging to the object will fix some of these omissions. In addition, we have to complete this process by adding two pixels around all the foreground objects from the disparity value of the object itself. By this, we gain edges with high resolution that affect the resolution of the image, as shown in Figure 6.10 and Figure 6.11 (e) and its close-up (f).

$$d_{Int}(x_{hole}) = \begin{cases} d_{Int}(x-1), & if \quad d_{Int}(x-1) \geq d_{Int}(x+1) \\ d_{Int}(x+1), & if \quad d_{Int}(x-1) < d_{Int}(x+1) \end{cases} \tag{6.9}$$

Clearly, omitting pixels that cause ghost contours in the step #4 leaves a clear corruption of these edges, making them appear like step-edges, as shown in Figure 6.4 (f); however, in this step, this corruption is enhanced by the proposed technique after extending the size of the foreground object from left and right sides.

$$d_{Int}(x+1) = d_{Int}(x+2) = d_{Int}(x), \qquad if \ d_{Int}(x) \ is \ edge \tag{6.10}$$

The effectiveness of the proposed filling technique is demonstrated in Figure 6.8. As the original disparity sometimes has an unknown area (black pixels) in it, as shown



(a) Original left disparity

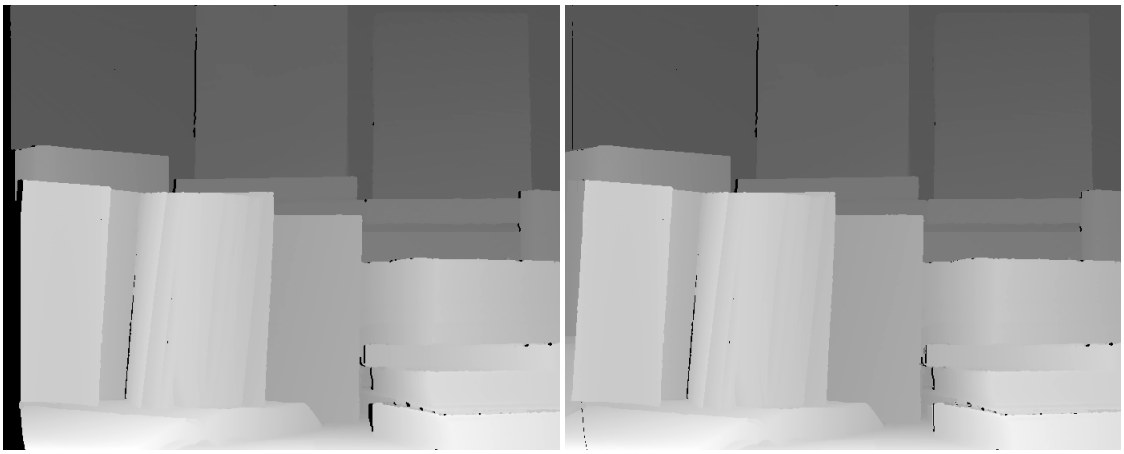
(b) Inpainted disparity with proposed algorithm

(c) Resulting holes and cracks in the intermediated disparity after compensating from left disparity

(d) After proposed hole filling techniques

(e) After filling disocclusion area

Figure 6.8: Depicting transferring the inpainted left disparity to the intermediate image process and the generated cracks and holes in it. Also depicting filling these holes by the proposed techniques.



(a) After filling holes with median filter (b) After filling disocclusion area

Figure 6.9: After filling the re-sampling holes with median filter and its effect after filling disocclusion area.

in Figure 6.8 (a), inpainting those areas is necessary before transferring their coordinates into the intermediate coordinates, as illustrated in Figure 6.8 (b). Now, the inpainting disparity has no holes in it, and we can clearly see the holes and cracks that were generated by mapping the coordinates of the inpainted original disparity onto the intermediate disparity, as shown in Figure 6.8 (c). Filling these holes and cracks by the proposed technique is demonstrated in Figure 6.8 (d), with the final disparity depicted in Figure 6.8 (e) after filling the disocclusion area. Hole-filling by median filter is demonstrated in Figure 6.9 (a), and we can see that the holes are not completely filled due to their size even though it is not a disocclusion area; this is proved in Figure 6.9 (b), where they are still unfilled.

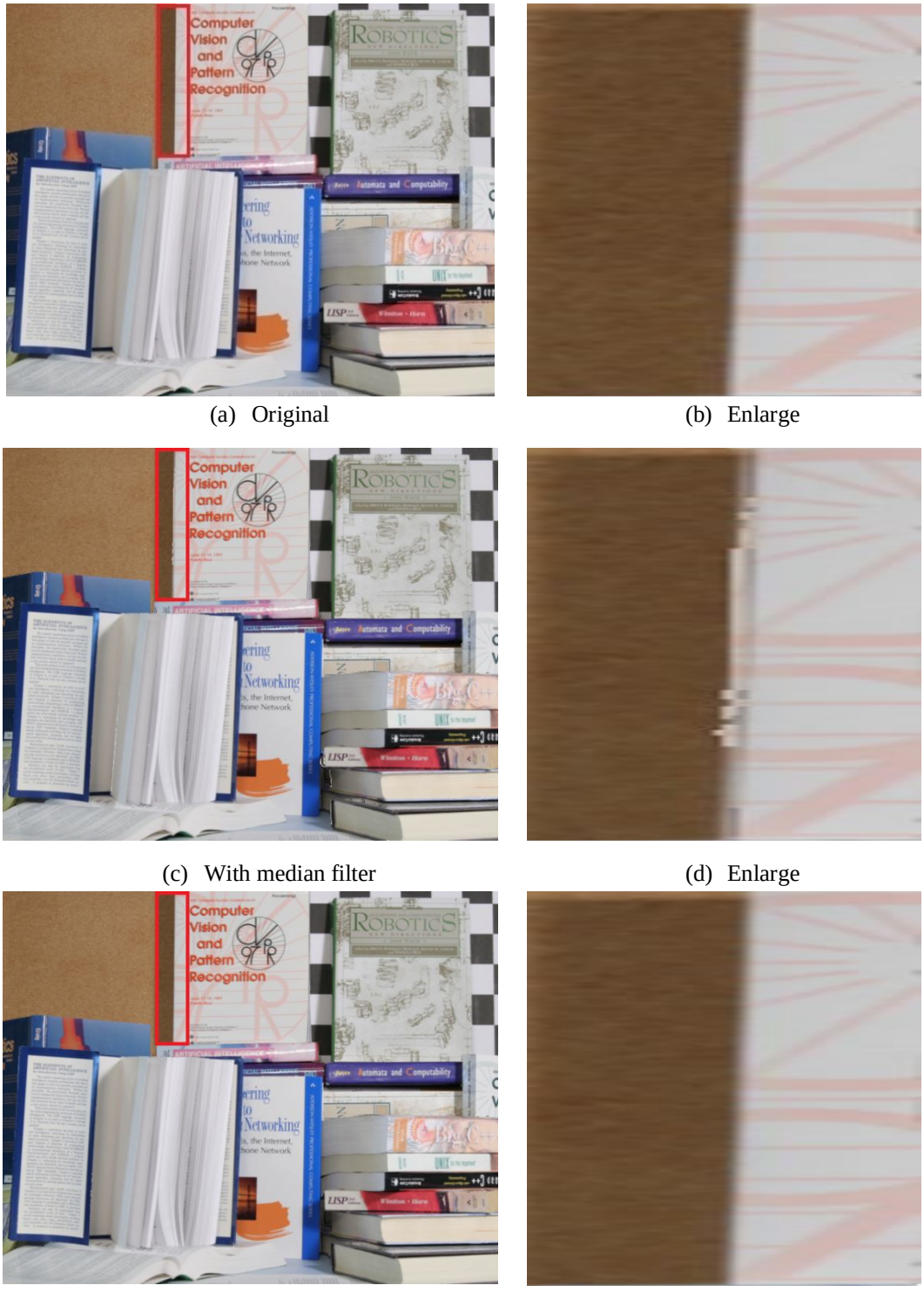
The comparison between the reconstructed image that is generated after filling the holes and cracks by median filter and by the proposed filling technique is shown in Figure 6.10, with the original intermediate image shown in Figure 6.10 (a). As we can see, edges are corrupted if the holes are filled by median filter, as shown in Figure 6.10 (d) which is an enlargement of a small chunk containing edges. The edges in Figure 6.10 (f) which were generated after filling by the proposed technique are identical to the edges of the original image in Figure 6.10 (b). Figure 6.11 is the same example of Figure 6.10 but using a different image.

(a) original (b) Enlarge

(c) With median filter (d) Enlarge

(e) With proposed technique (f) Enlarge

Figure 6.10: Comparing the reconstructed image using our proposed framework if we used the median filter to fill the re-sampling holes and the proposed techniques.



(a) Original (b) Enlarge

(c) With median filter (d) Enlarge

(e) With proposed technique (f) Enlarge

Figure 6.11: Comparing the reconstructed image using our proposed framework if we used the median filter to fill the re-sampling holes and the proposed techniques.

***Step #7*: Disocclusion handling**

Since all occluded and newly-exposed areas are inpainted in the d_L and d_R, we can recover the newly-exposed area in the d_{Int} from one of the known disparities (d_L and d_R). This is because the newly-exposed area in the d_{Int} is considered part of the newly-exposed area of the known disparities. Since, in our proposed reconstruction algorithm, the d_{Int} is built from d_L, the newly-exposed area in d_{Int} will be recovered from d_R accurately using the following equation:

$$d_{Int}(x) = d_R(x - (1-\alpha)d_L(x)) \ldots \quad if\ VM\ (x) = 1 \quad \forall x \in S_{Int} \tag{6.11}$$

***Step #8*: Estimate the texture of the intermediate image**

After generating an error-free disparity map for the intermediate image, we can reconstruct the texture of the intermediate image from surrounding known images. The reconstruction process depends on the visibility of each image point in the intermediate image, so we will use the visibility matrix built in step #3. Image points that are visible in one of the surrounding images will be reconstructed from this image, while we use the weighted-average for the image points that are visible in both images, as illustrated in equation (6.12). Also, using one image to reconstruct those points is another suggested solution that can be used.

$$I_{Int}(x) = \begin{cases} I_L(x + \alpha d_{Int}(x)) & VM\ (x) = 1 \\ I_R(x - (1-\alpha)d_{Int}(x)) & VM\ (x) = -1 \\ (1-\alpha)\ I_L(x + \alpha d_{Int}(x)) + \alpha\ I_R(x - (1-\alpha)d_{Int}(x)) & VM\ (x) = 0 \end{cases} \tag{6.12}$$

x belongs to sampling grid S_{Int}, but $x + \alpha\ d_{Int}\ (x)$ and $x - (1 - \alpha)\ d_{Int}\ (x)$ may not belong to S_L and S_R. However, this can be solved easily using spatial interpolation as follows:

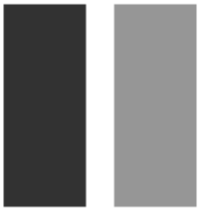
$$L_{pos} = x + \alpha d_{Int}(x) \tag{6.13}$$

$$I_L(L_{pos}) = (\lceil L_{pos} \rceil - L_{pos}) * I_L(\lfloor L_{pos} \rfloor) - (L_{pos} - \lfloor L_{pos} \rfloor) * I_L(\lceil L_{pos} \rceil) \tag{6.14}$$

Equation (6.14) is also applied on R_{pos} of $x - (1 - \alpha) * d_{Int}(x)$. Thus, we generate an intermediate image with filled holes, recovered disocclusion regions, removed ghost contours, and high resolution for edges.

***Step #9*: Inpaint double-occlusion area in the intermediate image**

The visibility matrix contains four labels (1,-1, 0, and 2), and we have handled the first three labels while the last one, which refers to double occlusion, has not yet been handled. In particular, these areas in d_{Int} are not visible in any of the surrounding images. Double occlusion areas do not often occur and, should one be found, it will be small in relation to the image size. Inpainting a double occluded area in d_{Int} leads to false reconstruction of those areas in the interpolated image due to the absence of these pixels from the reference images. Therefore, separating inpainting as a post-processing for the interpolated image is a suitable solution to recover those areas from the neighbouring pixels.



(a) Left image

(b) Right image

(c) Interpolated image with double occlusion area

(d) After inpainting double occlusion area

Figure 6.12: Showing the double-occlusion area problem before and after inpainting

As the inpainting will be applied to remove a small object from the texture images, many algorithms have been proposed for this purpose. So, we inpaint the double occluded area in our reconstruction algorithm by using the Exemplar-based inpainting algorithm [138], which combines the advantages of texture synthesis algorithms and inpainting techniques. Synthesized pixel values are diffused in a similar way to the diffusion of pixels in inpainting, which results in inpainting both texture and structure. Figure 5.12 shows an example of how a double occlusion area appears in a reconstructed image and demonstrates how the Exemplar-based inpainting technique inpaints those areas accurately.

6.4 Summary

In this chapter, a novel intermediate view reconstruction algorithm has been proposed to overcome the deficiencies of the previous reconstruction algorithms. This novel algorithm focuses on how to find an intermediate image with a given disparity between two input images. As the detection and handling occlusion problem are considered from the challenges of the intermediate view reconstruction, we employed the algorithms proposed in Chapters 4 and 5 in a novel occlusion-aware intermediate view reconstruction. We proposed three quality improvement techniques to reduce image artifices. First, instead of using a median filter to fill cracks and holes produced by image re-sampling, we proposed an object-based filling technique that fills, by the object's intensity, the holes located in a foreground object or around its edges. Subjective and objective evaluations proved that the proposed technique produces more accurate results around object boundaries. Second, we removed the ghost contours from the virtual view and restored them to their correct place by extending the newly-exposed area three pixels in the direction where the ghost contour is located. Third, since the novel reconstruction algorithm detects and inpaints the occlusion areas in the two input disparity maps, the disocclusion areas are recovered from one of the input images. Inpainting a double-occlusion area as a post-processing step is considered a reasonable solution to this problem since it is not visible in both reference images, so inpainting in the disparity maps is not a viable solution.

CHAPTER

7

Occlusion-Aware Intermediate View Reconstruction Algorithm Results

An evaluation of our proposed reconstruction algorithm is presented in this chapter. We compare the proposed reconstruction algorithm to the standard backward- and forward-projection approaches quantitatively using PSNR, and qualitatively by conducting a human-trial experiment on a 3D display.

7.1 Experiment

7.1.1 Objective

In order to evaluate the proposed reconstruction algorithm qualitatively on a 3D display, a subjective human trial is performed to assess the quality of the reconstructed intermediate image on 3D display relative to the reference intermediate image based on the ITU-R 500 recommendation and comparing it to the reconstructed images by backward and forward algorithms. ITU-R 500 recommendation is a methodology that followed to assess the quality of television pictures by number of participants [139]. In section 7.1.3, we will explain in details the followed methodology of this recommendation.

7.1.2 Hypothesis

Our expectation is that the perceived image quality of the proposed reconstructed algorithm from the subjective human-based trial will be better than the reconstructed images by backward and forward algorithms. Thus, the difference between the reference image and the reconstructed image obtained by the proposed algorithm will be smaller than the difference between the reference image and the one reconstructed by the existing algorithms.

7.1.3 Method

In this experiment, we followed the double-stimulus continuous quality scale (DSCQS) method for stereoscopic image based on the ITU-R 500 recommendation [139]. Using the DSCQS method, the participants view a pair of stereoscopic images of the same image; one is the reconstructed (degraded) image and the other one is the reference image. The participants are asked to evaluate the quality of both images. Based on the ITU-R 500 recommendation, each participant has a session of no longer than 30 minutes' duration. A sequence of stereoscopic image pairs of different types is presented to the observers in random order. At the end of the experiment, the mean of difference scores of all participants for each algorithm per image are computed.

7.1.4 Equipment and Viewing Conditions

A full-resolution True 3Di stereoscopic display is used for displaying 3D experimental test images. This display consists of a twin-LCD display, one for the left eye and the other for the right eye; each one is a 24-inch with 1920 × 1200 resolution. Thus, a full-resolution view is delivered to each of the viewer's eyes. The participants are required to wear polarized glasses to combine the left and right images on a virtual 3D image. The grading scores are run separately on a 15.4-inch Toshiba LCD monitor with 1280 × 800 resolution. In this experiment a Dell Precision PWS670 computer with NVIDIA Quadro FX5600 graphics card is used. The experiment is carried out in a lit room with organized equipment as shown in Figure 7.1.

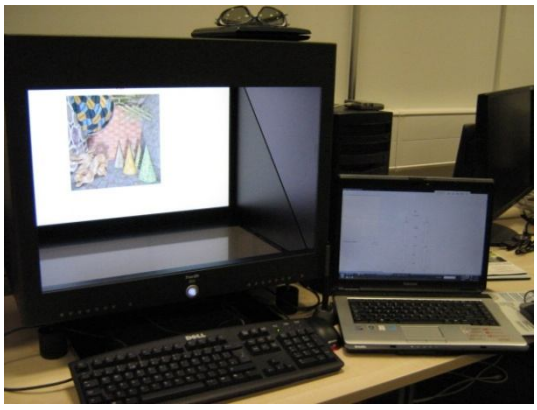


Figure 7.1: The equipments used in the experiment

7.1.5 Test Images

The experiment is performed on two types of images - real and synthetic - and there are thirteen for each type. The real images are downloaded from the Middlebury site, while 8 of the synthetic images are computer-generated and 5 are downloaded from the Alberta University dataset. Each is assessed with three different algorithms to show the participant 39 real images and 39 synthetic images. In each used dataset, we select two images as reference images and from those we reconstruct two virtual images in between them to be tested in the subjective assessment. Thus, the participant will see an interpolated image in each eye so the stereoscopic image will be fully interpolated; therefore, our reconstruction algorithm will be scored accurately, as demonstrated in Figure 7.2. This is in contrast to the performed experiments in the literature [101, 103], which show the participants the reference image in one eye and the interpolated image in the other; these will be evaluated inaccurately.

7.1.6 Participants:

In total, 21 subjects (13 male, 8 female) participated in this experiment. Their ages range from 22 to 38, with a mean of 28 years. Subjects who meet minimum criteria in three vision tests are selected as assessors. Following ITU-R 500 recommendations, all participants are non-experts in assessing image quality in their normal work. In addition, they are not aware that one of the images is a reference image.

7.1.7 Protocol:

7.1.7.1 Procedure

The experiment is divided into three phases: vision test phase, training phase and trial phase. Firstly, in a vision test phase three quick tests are conducted on the participants; only those participants who meet the minimum criteria for vision acuity with 20:30, stereo-acuity at 40 sec-arc, and who passed an Ishihara colour vision test will participate in the experiment. After that, the selected assessors are trained on five pairs of stereoscopic images in a training phase. They are asked to assess the quality of the trial images as part of the assessment procedure, although the results of

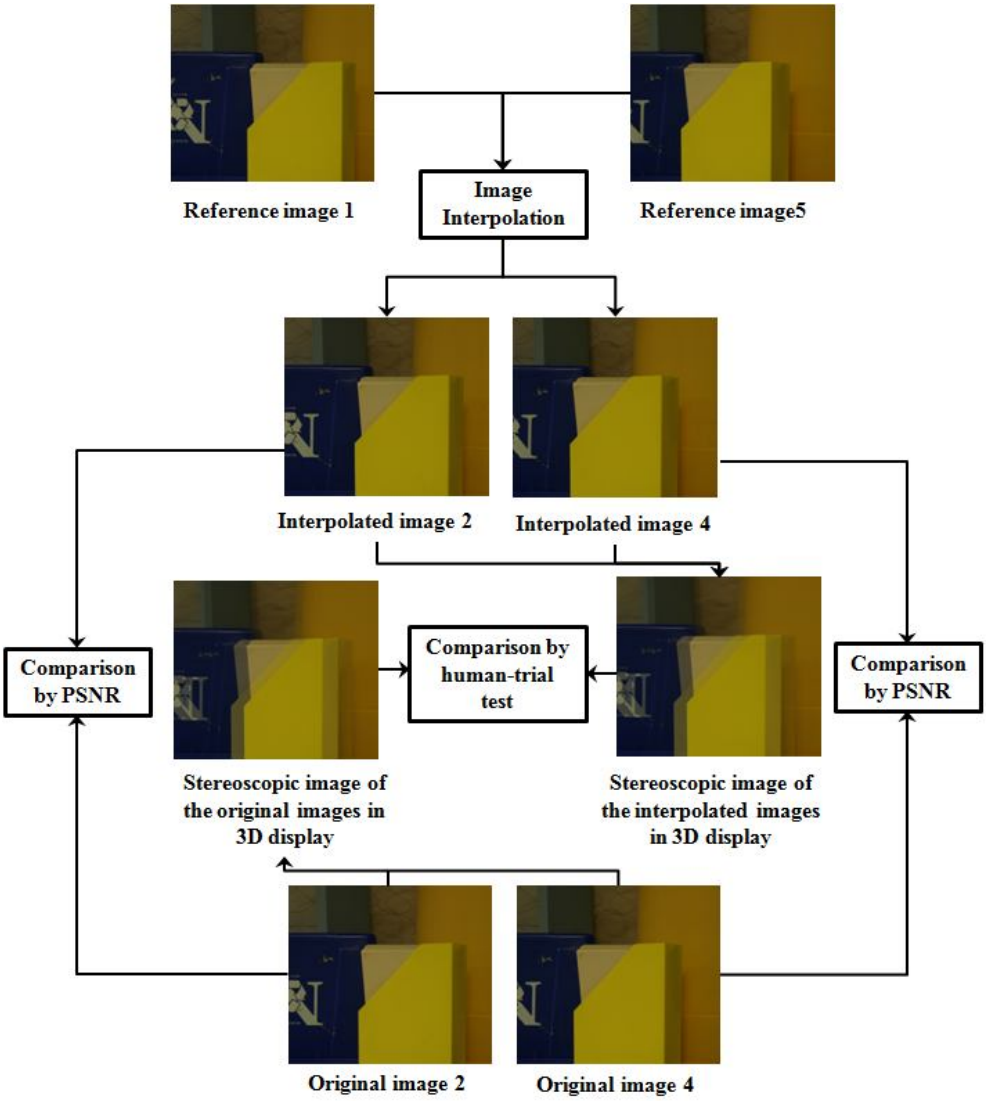


Figure 7.2: Depicts the overall system of the experiment

the training phase will not be included in the trial results analysis. Then, the participant is ready to start the experiment trial to assess 84 experimental images in two sessions with a short break in between. At the beginning of each session, three images are added to stabilize the participant's opinion although they are not included in the results analysis as they are repeated randomly in each session.

Each participant views these images in a different order from the other participants. Each time, the participants see two images and are advised to switch between them up to 4 times even though no restrictions are enforced. Scoring the quality of these two images is performed in a different 2D display where the participants are asked to

adjust the slider bar to indicate the assessed quality of the images. Once the participants have submitted their results, the next pair of stereoscopic images is displayed in a 3D display and they cannot go back to the previous images.

In total, the experiment takes half an hour, as the three vision tests and the training session take about 15 minutes, while the trial session takes another 15 minutes including a short break in the middle. All participants are aware that they can withdraw from the trial at any time and they are given the chance to ask questions at any time during the experiment.

7.1.7.2 Grading Scales

As the participants view two stereo images simultaneously, two sliding bars, one for each image, are displayed on the 2D display, as shown in Figure 7.3. For each pair of stereo images, the participants are asked to assess the quality of each stereo image by choosing the appropriate scale from the following five scales: Excellent, Good, Fair, Poor, and Bad. Based on the normal ITU-R five-point quality scale, these scales provide a continuous rating system although they are divided for guidance purposes. Once the participant has moved the sliding bar to the desired scale and pressed submit, the assessment is converted to a normalized score between 0 and 100 and recorded on the scoring sheet.

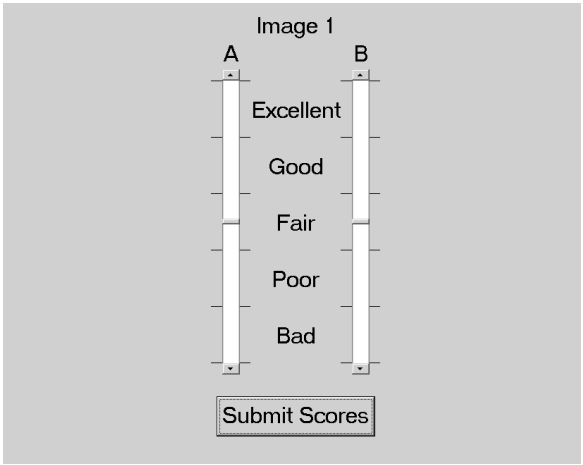


Figure 7.3 Grading score of a pair of images

7.2 Results and Analysis

An evaluation of our proposed reconstruction algorithm is presented in this section. Based on the materials used and the experiments conducted, we will discuss the result analysis, the measurements and a comparison of the results for our proposed algorithm with the standard techniques of this field in the subsequent sections.

7.2.1 Materials

The performance of our proposed reconstruction algorithm is tested on various datasets. The real images are downloaded from the Middlebury site and selected from different datasets; these images are depicted in Figure A.1 in Appendix A. Some of the datasets contain nine images while the others contain seven images. Specifically, from the nine-image dataset, we choose image 2 and image 6 as a reference images due to the availability of disparity maps for them, while image 3 and image 5 are considered as original virtual images. From the seven-image dataset, image 1 and image 5 are chosen as reference images and image 2 and image 4 as original virtual images. The Middlebury datasets are rectified images obtained from cameras arranged in parallel setup geometry. They are associated with high-quality disparity maps but they contain unknown areas that need inpainting.

As for the synthetic datasets, eight of them are generated by graphics software and five are downloaded from Alberta University that are shown in Figure A.2 and Figure A.3, respectively, in Appendix A. For each generated dataset we assumed the distance between the reference images is a normalized distance, so the I_L is located at $\alpha = 0$, and the I_R at $\alpha = 1$. Thus, the reconstructed images should be at $0 < \alpha < 1$, so we generate two original images for the reconstructed images at $\alpha = 0.25$ and $\alpha = 0.75$ to be in the same position as the real images. The disparity maps are generated accurately by giving a depth for the object while leaving the background static.

The Alberta University dataset consists of three different datasets, each one containing four different images (left image and right image with three different baseline separations) with ten different textures and three different amounts of noise. We select the leftmost and the rightmost (at 45 pixel baseline) images as reference

images; the right images at a baseline of 15 and 30 pixels are considered original virtual images.

7.2.2 Experimental Results

The reconstructed images from our proposed reconstruction algorithm are evaluated quantitatively using PSNR metric and qualitatively by human trial experiment. Although the PSNR is a good metric to compare images, it does not indicate how high the quality of the reconstructed images will be in the 3D display. Therefore, we conduct a human trial test empirically to measure whether the reconstructed image is of good quality in the 3D display or not.

7.2.3 PSNR

Peak Signal to Noise Ratio (PSNR) is calculated by comparing the interpolated image with the original one to compute the Mean Square Error (MSE) which results from the corrupted noise. The MSE and PSNR equations are calculated as in equation (5.21) and (5.22) in Chapter 5, where $I(x,y)$ is the original interpolated image, $I'(x,y)$ is the interpolated image, M and N are the dimensions of the images and max_p is the maximum pixel value in the image. Higher PSNR refers to higher image quality due to less error noise. To calculate the PSNR for the coloured (RGB) images, it is converted to the YCbCr space first. Then, the PSNR is calculated for the Y (luma) component since it contains a weighted average for the R, G, and B components, giving the G the highest weight.

7.2.3.1 PSNR results

For each different algorithm, the PSNR is calculated for the real and synthetic image type as shown in Tables 7.1 and 7.2 and drawn in Figure 7.4 and Figure 7.5, respectively. Also, the average of PSNR for 13 images of each algorithm under two image types is depicted in Table 7.3. As we can see, the proposed algorithm gained on average a higher PSNR of ~9 db than a standard backward algorithm and ~14 db than a standard forward algorithm. However, the following results from ANOVA illustrate these results.

Table 7.1: The PSNR of 13 real images for three different algorithms

Image Name	Backward	Forward	Novel
R1	28.42	17.87	37.02
R2	22.81	20.43	36.05
R3	26.73	20.42	35.38
R4	27.78	21.31	35.96
R5	27.25	23.31	36.11
R6	23.97	20.61	41.52
R7	28.02	24.73	33.98
R8	33.05	27.19	40.26
R9	26.18	20.22	39.82
R10	27.11	22.48	38.80
R11	31.91	25.02	38.26
R12	30.24	25.86	33.63
R13	28.66	25.52	31.77

Table 7.2: The PSNR of 13 synthetic images for three different algorithms

Image Number	Backward	Forward	Novel
S1	30.42	40.17	49.79
S2	24.26	27.06	38.29
S3	27.09	29.56	39.90
S4	26.13	29.46	37.74
S5	27.99	30.92	47.28
S6	25.67	29.76	35.89
S7	22.36	24.00	43.24
S8	24.96	24.03	43.26
S9	29.94	24.27	39.67
S10	28.30	22.23	40.03
S11	29.01	21.70	40.23
S12	29.01	23.67	40.62
S13	27.91	22.25	38.60

7.2.3.2 ANOVA analysis for PSNR results

Two-way ANOVA is conducted to see whether there is a significant difference between the three different reconstruction algorithms, the two types of image, and their interaction in the mean of difference scores. To investigate which of these algorithms are significantly different, each pair of means is compared by Tukey Multiple Comparison Test.

The results from two-way ANOVA indicated that the three different algorithms are statistically significant, (p value = 0.000), with a 0% probability of them being the same. Also, there is a significance difference between the two types of images (real and synthetic), (p value = 0.001) < 0.05. Moreover, the interaction between these two variables (Image type * Algorithm) is statistically significant (p value = 0.002.)

The result from the Tukey multiple comparison test showed that there is no significance difference between Backward and Forward algorithms as p value is 0.426, while the Novel algorithm is 100% significantly different from the other algorithms. This is clearly shown in Figure 7.4 and Figure 7.5.

Table 7.3: The mean and standard deviation of PSNR for three different algorithms under two type of images. (N refers to the number of images)

Image Type	Algorithm	Mean	Std. Deviation	N
Real	Backward	27.8562	2.82337	13
	Forward	22.6900	2.80106	13
	Novel	36.8123	2.83742	13
Synthetic	Backward	27.1570	2.36924	13
	Forward	26.8521	5.15252	13
	Novel	41.1187	3.88443	13

Figure 7.4: PSNR of the 13 real images for three different algorithms

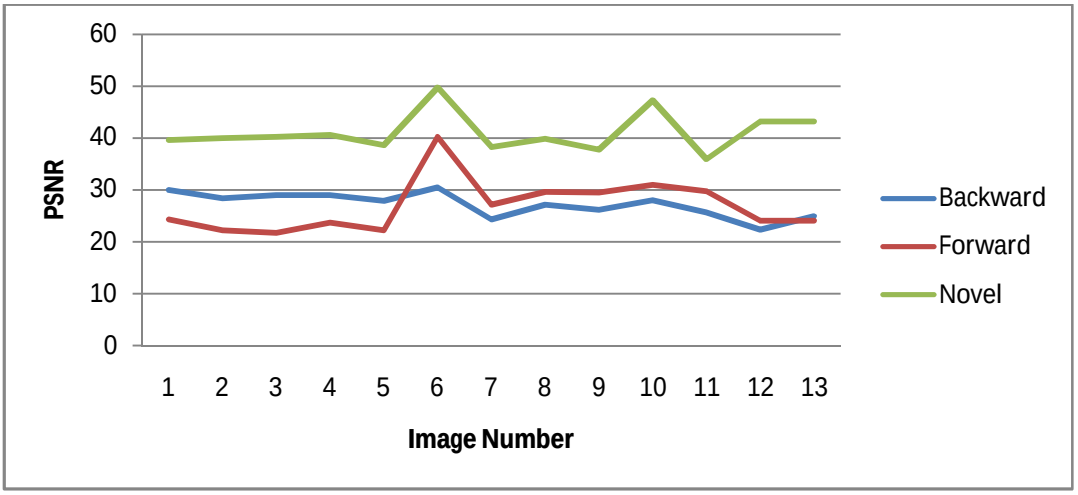


Figure 7.5: The PSNR of 13 synthetic images for three different algorithms

7.2.3.3 Effectiveness of the proposed hole-filling technique vs. median filter:

Instead of using the proposed hole-filling technique, a median filter of size 3 × 3 is used to fill the holes and cracks in the virtual disparity to reconstruct a virtual image. Then, a PSNR is calculated for 16 images which are then compared to the PSNR of our reconstructed images by using the proposed hole-filling technique. These PSNR are depicted in Table 7.4. The quantitative measures show the effectiveness of the proposed hole-filling technique, which gained 1.22 db over a median filter.

Table 7.4: The PSNR of 16 images (13 real, 3 synthetic) which show the effectiveness of the proposed hole-filling technique over a median filter.

Image Name	With median filter	With proposed object-based technique
R1	36.17	37.02
R2	34.20	36.05
R3	34.17	35.38
R4	35.70	35.96
R5	35.37	36.11
R6	40.65	41.52
R7	40.20	40.26
R8	33.01	33.98
R9	37.72	39.82
R10	37.38	38.80
R11	37.05	38.26
R12	33.42	33.63
R13	30.80	31.77
S11	38.09	40.03
S12	37.37	40.62
S13	36.98	38.60

7.2.3.4 Effectiveness of the linear interpolation vs. nonlinear interpolation

The texture of the intermediate image can be reconstructed based on the visibility of each image point in the intermediate image. So image points that are visible in one of the surrounding images will be reconstructed from this image. On the other hand, linear and non-linear interpolation can be used to reconstruct image points that are visible in all the reference images. In a linear interpolation, a weighted-averaging of all the available reference images is used to estimate the texture of the image points that suffer from a blurring effect while, in non-linear interpolation, one of the reference images that suffer from a patchiness effect is used. Therefore, the PSNR is calculated for 20 images (15 real, 5 synthetic) to demonstrate which of the above methods gives a result close to the original intermediate image. These PSNR are depicted in Table 7.5. The quantitative measures show the effectiveness of the linear interpolation, which gained 1.21 db over a non-linear interpolation.

Table 7.5: The PSNR of 20 images (15 real, 5 synthetic) showing the effectiveness of the linear interpolation vs. non-linear interpolation for estimating the texture of intermediate image points that are visible in all the reference images.

Image Number	PSNR with linear interpolation	PSNR with nonlinear interpolation	Difference
Barn2 (R11)	38.26	37.37	0.89
Bull (R7)	40.26	39.82	0.44
Poster (R13)	31.77	30.80	0.97
Sawtooth (R8)	33.98	33.00	0.98
Venus (R12)	33.63	32.98	0.65
Book (R2)	36.05	35.04	1.01
Bowling (R5)	36.11	35.58	0.53
Doll (R3)	35.38	34.75	0.63
Lampshade (R1)	37.02	36.23	0.79
Midd1 (R4)	35.96	35.35	0.61
Plastic (R6)	41.52	40.38	1.14
Reindeer	32.61	32.18	0.43
Rock (R10)	38.80	37.50	1.30
Teddy	33.22	32.80	0.42
Wood (R9)	39.82	39.38	0.44
S11	40.62	38.92	1.70
S12	41.04	38.27	2.77
S13	40.11	36.90	3.21
S14	41.52	39.18	2.34
S15	39.22	36.26	2.96

7.2.3.5 Effectiveness of the proposed inpainting technique vs. other methods:

Due to the availability of ground truth disparity, the occlusion areas do not need inpainting; therefore the disocclusion areas in the virtual disparity are recovered from one of the available disparities. The above PSNR is calculated for this disocclusion recovery. Thus, the occluded areas are detected and added to the disparity for inpainting by the proposed inpainting technique and then an intermediate image is reconstructed by the proposed reconstruction algorithm. The PSNR for these reconstructed images is calculated to compare it to the PSNR of reconstructed images using ground truth disparity. This comparison is depicted in Table 7.6. As we can see, the PSNR values are very close to one another; even the occlusion areas are very large and inpainting them is considered a hard task.

Table 7.6: Comparing the PSNR of the reconstructed images using ground truth disparity vs. the reconstructed images using the inpainted disparity by the proposed inpainting technique.

Image Number	PSNR	With occlusion inpainting
barn2 (R11)	38.2560	38.2401
Bull (R7)	40.2638	40.2299
Plastic (R6)	41.5169	40.9369
Poster (R13)	31.7716	31.7311
Sawtooth (R8)	33.9812	33.8752
Venus (R12)	33.6314	33.6277
Wood (R9)	39.8229	38.8951
Teddy	33.2245	33.1645
Reindeer	32.6078	32.0756

As we have compared, in chapter 5 the proposed inpainting technique to the other four techniques (HEDI, ROIFILL, FOE, and Exemplar-based), it is necessary to compare the effectiveness of these inpainting techniques on the reconstructed images. Therefore, in Table 7.7 we have compared the PSNR of the reconstructed image using inpainted disparity by the proposed inpainting technique with the reconstructed images using the inpainted disparity by other inpainting techniques. We conclude that the PSNR using the proposed inpainting technique is relatively close to the original disparity even though this disparity has unknown areas that are also inpainted explicitly by the proposed inpainting technique. Moreover, the occluded area of the Reindeer disparity image that is shown in Figure 5.14 is considered a complicated area where a large part of the arc has disappeared, and inpainting it is considered a hard task. Therefore, this close PSNR is a positive evaluation of our proposed inpainting technique.

Noticeably, the PSNR using the proposed inpainting technique is higher than the other four techniques. ROIFILL technique has the lowest PSNR due to incompleteness of the filling process for the occlusion areas. As we can see, the image size for FOE and Exemplar-based inpainting techniques is (447 × 370) which is smaller than that used for other inpainting techniques (671 × 555), because a larger image cannot be completely inpainted due to the out-of-memory problem.

Table 7.7: Compares the PSNR of four different inpainting techniques with the PSNR of the proposed inpainting technique.

Inpainting Technique	PSNR	Image size	Inpainting time
Original[6]	32.6078	671 × 555	-
proposed	32.0756	671 × 555	19.448148
HEDI	31.4672	671 × 555	0.413368
ROIFILL	24.4828	671 × 555	1.380815
FOE	29.2865	447 × 370	304.611208
Exemplar	28.6536	447 × 370	434.515383

7.2.3.6 Step-by-step quality improvement

To measure the quality improvement for each step in the intermediate view reconstruction algorithm, the PSNR is calculated after each quality improvement step and the values are drawn in one figure as shown in Figure 7.6. The PSNR for each improvement step is depicted in Table 7.8. Noticeably, each quality improvement step affects the improvement procedure. Specifically, the disocclusion handling step shows a major improvement due to the large size of the disocclusion area relative to the other area processed by other improvement steps.

Table 7.8: Step-by-step PSNR for the proposed improvement techniques in the proposed reconstruction algorithm.

Processing Steps	PSNR
Basic backward	25.7648
With inpainting unknown areas	26.3270
Handling visibility problem	27.8303
Solving ghost contours	28.1642
Hole and cracks Filling	29.3914
Disocclusion handling	33.2245

[6] Original refers to the original disparity with unknown area that inpainted by the proposed inpainting algorithm also.

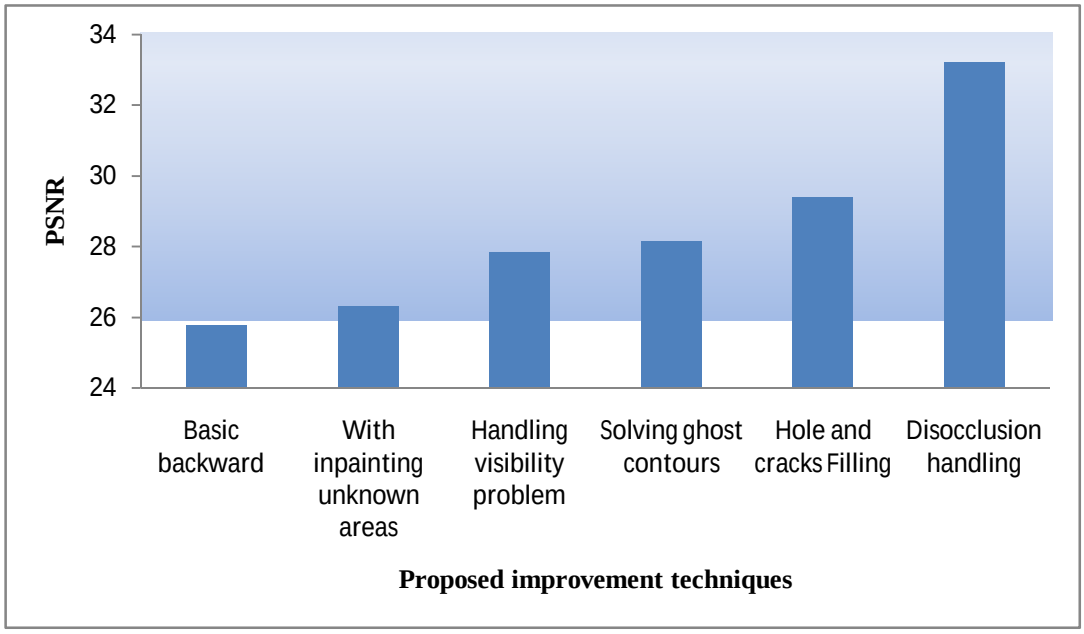
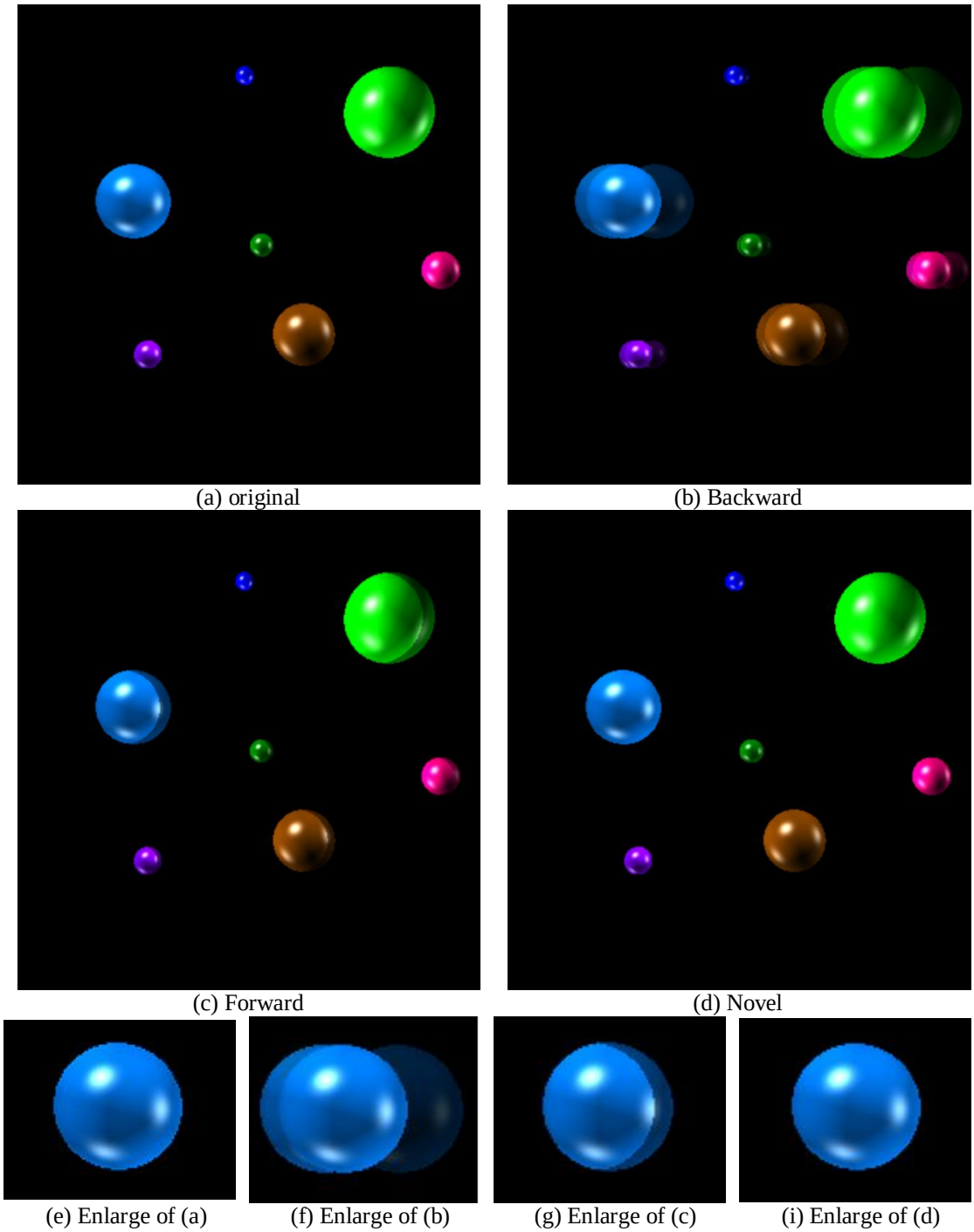


Figure 7.6: Showing step-by-step quantitative measures for the proposed improvement techniques

7.2.3.7 Examples on reconstructed images

Examples of reconstructed images by three different algorithms are demonstrated in Figure 7.7, Figure 7.8, and Figure 7.9. In each figure, a different image type is used. Figure 7.7 is an example of a synthetic image generated by computer graphics software and is considered a very simple image. Figure 7.8 is another example of a synthetic image but it is downloaded from Alberta University and is more complicated than the former. Figure 7.9 is an example of a real image downloaded from the Middlebury site.

These figures clearly show that our proposed algorithm reconstructs an intermediate image which is very close to the original virtual image. On the other hand, the reconstructed images using backward and forward algorithms have many corruption areas. With the backward algorithm, the main corruption areas are located in the occlusion area because it is not handled, while the reconstructed image by forward algorithm has many cracks on it due to the generation of undefined points in the final image, in addition to the corruption of the occlusion areas.



(a) original (b) Backward

(c) Forward (d) Novel

(e) Enlarge of (a) (f) Enlarge of (b) (g) Enlarge of (c) (i) Enlarge of (d)

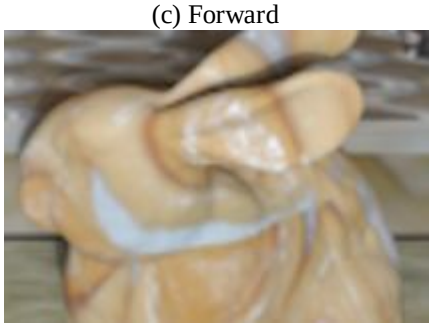
Figure 7.7: showing the reconstructed image from the three different algorithms and comparing them to the original

(a) original

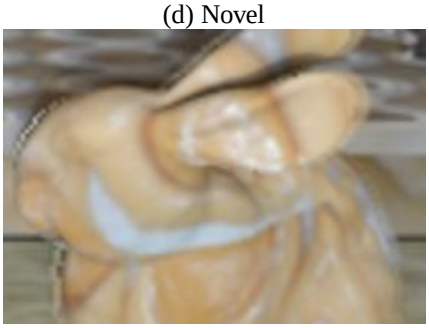
(b) Backward

(c) Forward

(d) Novel



(e) Enlarge of (a)

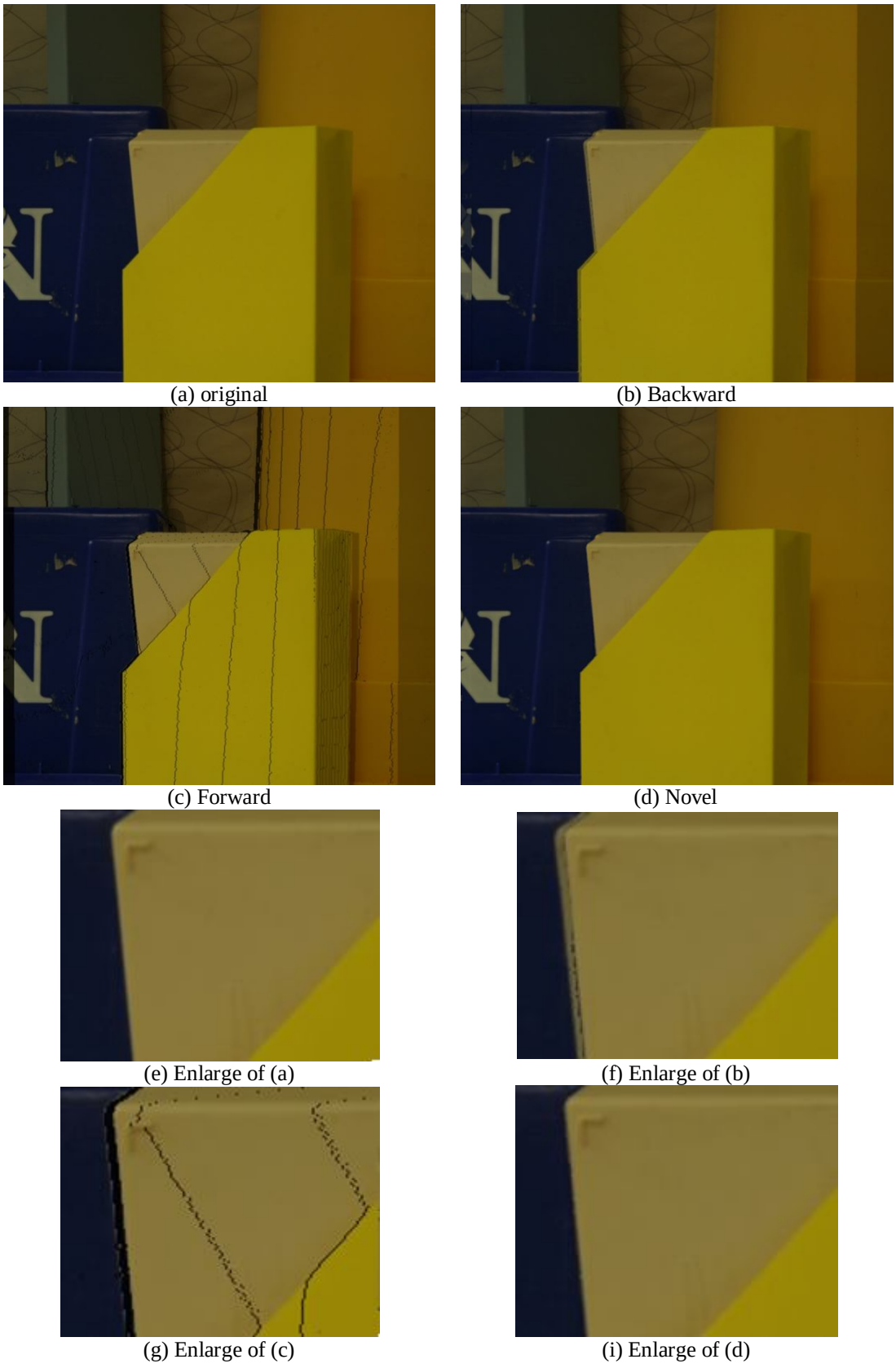


(f) Enlarge of (b)

(g) Enlarge of (c)

(i) Enlarge of (d)

Figure 7.8: showing the reconstructed image from the three different algorithms and comparing them to the original



(a) original (b) Backward

(c) Forward (d) Novel

(e) Enlarge of (a) (f) Enlarge of (b)

(g) Enlarge of (c) (i) Enlarge of (d)

Figure 7.9: showing the reconstructed image from the three different algorithms and comparing them to the original

7.2.4 Subjective Human Trial

Data collected from the subjective human trial are subjected to two-way Analysis of Variance (ANOVA), where the reconstruction algorithm and image type are considered as independent variables and scores as dependent variables. The confidence interval is assumed to be 95%. Two-way ANOVA is conducted to see whether there is a significant difference between the three different reconstruction algorithms, the two types of image, and their interaction in the mean of the difference scores. To investigate which of these algorithms are significantly different, each pair of means is compared by Tukey Multiple Comparison Test.

As the assessment scores will be normalized scores in the range [0,100], the difference between the assessment of the reference and the reconstructed image sequences will be calculated for each image. Greater difference indicates worse perceived image quality. Figure 7.10 and Figure 7.11 show the box plots of the results for all participants for algorithm and image type independent variables, respectively. The distribution of the mean of difference score for the three different algorithms is depicted in Figure 7.12.

The box plot in Figure 7.10 and the distribution in Figure 7.12 indicate that the proposed algorithm has smaller differences i.e. smaller mean scores, than the backward and forward algorithms. Specifically, it is very clear from Figure 7.12 (c) that the mean of difference score for the proposed algorithm is located between 3 and 30, which yields a small difference between the reconstructed image and the original one. On the other hand, the mean of difference scores for the generated image using backward and forward algorithms are distributed between 29 and 54, and 30 and 61, respectively, which yield a greater difference between the reconstructed images and the original one. Therefore, the quality of the reconstructed image using the proposed algorithm is closer to the original image than the reconstructed images by backward and forward algorithms. In addition, the mean of difference scores and standard deviation for each algorithm under two types of image is shown in Table 7.9. As we notice in this table, there is a big gap between the overall mean of difference scores of the proposed algorithm and the backward and forward algorithms under two types

of images. The following ANOVA analysis determines whether the proposed algorithm is statistically different from the other algorithms or not.

Table 7.9: The mean of difference scores and standard deviation for each algorithm under two type of images

Image Type	Algorithm	Mean	Std. Deviation	N
Real	Backward	41.7875	6.39614	13
	Forward	51.8168	6.01227	13
	Novel	15.8352	9.80443	13
Synthetic	Backward	45.9670	4.97455	13
	Forward	46.9817	7.96595	13
	Novel	10.8242	5.94318	13

The results from two-way ANOVA indicated that the three different algorithms are statistically significant, (p value = 0.000), with a 0% probability of them being the same. There is no significant difference between the two types of images (real and synthetic), (p value = 0.239) > 0.05. At the same time, the interaction between these two variables (Image type * Algorithm) is statistically significant (p value = 0.031.)

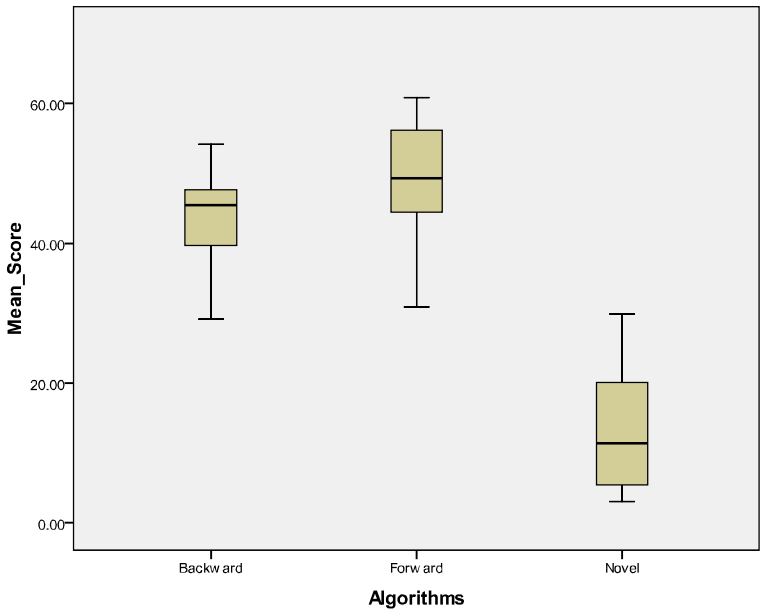


Figure 7.10: Box Plot results for three different algorithms

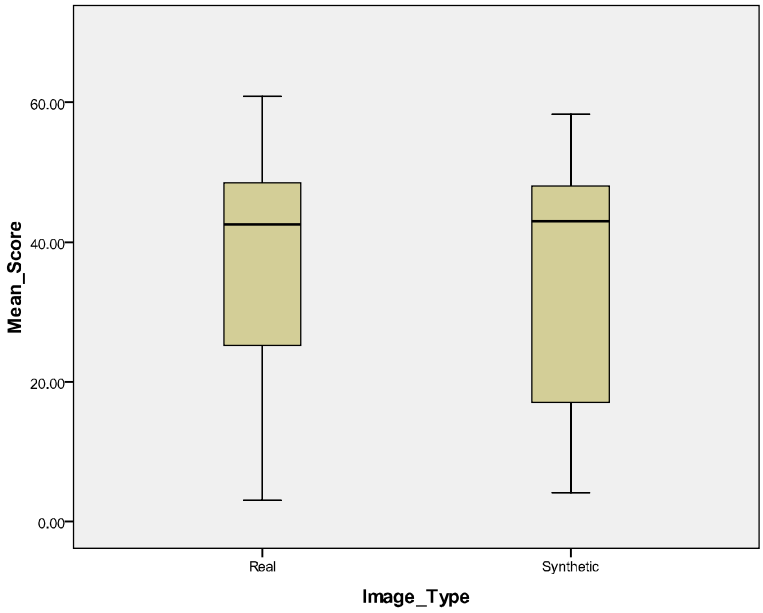


Figure 7.11: Box Plot results for two types of images

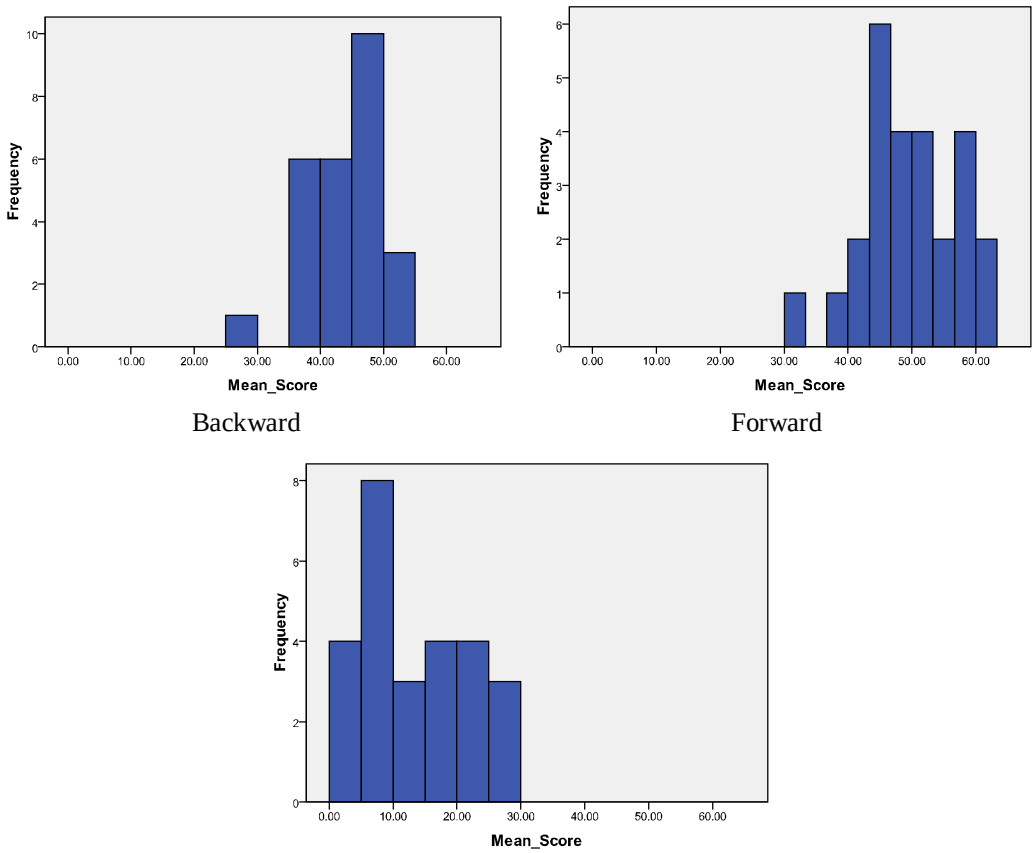


Figure 7.12: Showing the distribution of the mean of difference scores of the three different algorithms

The result from Tukey multiple comparison test showed that there is a significant difference between Backward and Forward algorithms as *p* value is 0.016, while the Novel algorithm is 100% significantly different from the other algorithms. This is clearly shown in Table 7.10. The mean and standard deviation of the difference score are depicted in Table 7.11, Figure 7.13 and Figure 7.14.

Table 7.12 shows that the mean of difference scores for the two image types are close to each other and this is evidence that there is no significant difference in the mean of difference scores. However, the interaction between image type and algorithm is statistically significant. This means that, under real image type as well as under synthetic image type, the mean of difference scores of the three different algorithms are statistically different. This is shown in Table 7.13. On the other hand, to verify whether the mean of difference score of each algorithm under two types of image (i.e. interaction between algorithms and image type) is statistically different or not, a t-test is conducted to investigate. Indeed, there is no significant difference between the mean of difference score of the novel algorithm when applied to the real and synthetic images (p value = 0.128). In addition, there is no significant difference using backward and forward algorithms with p values = 0.075 and 0.093, respectively. Table 7.14 clearly shows the overall mean scores of the interaction between algorithms and image types (algorithms * image type). Figure 7.15, Figure 7.16, and Figure 7.17 show the mean scores of each image type for novel, backward, and forward algorithms, where the real and the synthetic mean scores are very close to each other.

Table 7.10: Tukey multiple comparison test results

(I) Algorithms	(J) Algorithms	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Backward	Forward	-5.5220*	1.95037	.016	-10.1895	-.8545
	Novel	30.5476*	1.95037	.000	25.8801	35.2151
Forward	Backward	5.5220*	1.95037	.016	.8545	10.1895
	Novel	36.0696*	1.95037	.000	31.4021	40.7371
Novel	Backward	-30.5476*	1.95037	.000	-35.2151	-25.8801
	Forward	-36.0696*	1.95037	.000	-40.7371	-31.4021

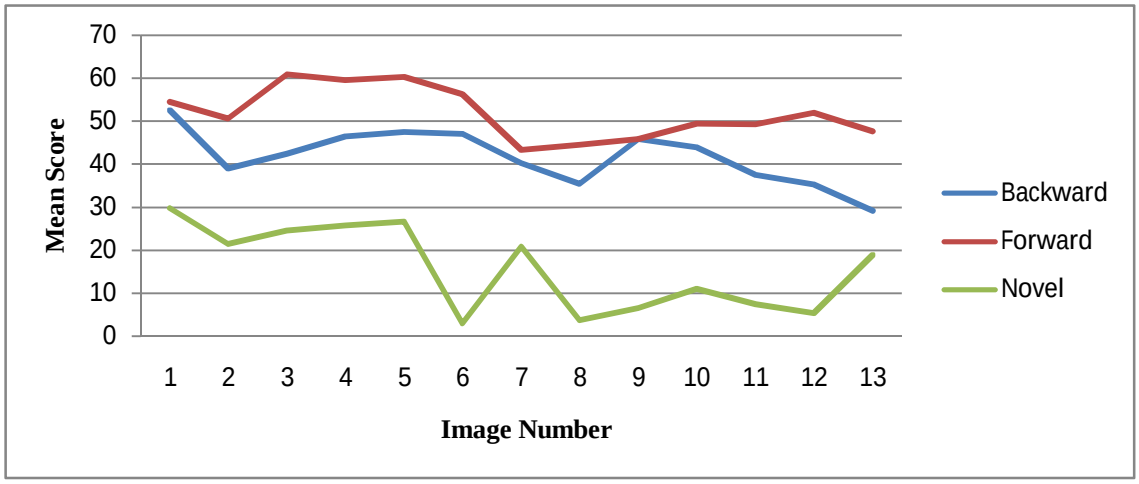


Figure 7.13: Mean of difference scores for three different algorithms for the real image type

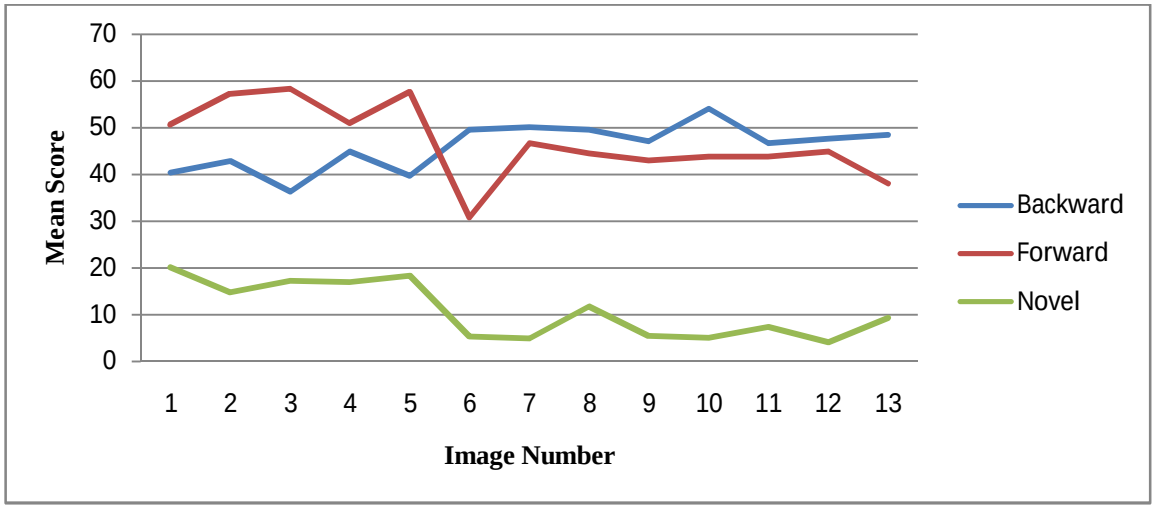


Figure 7.14: Mean of difference scores for three different algorithms for synthetic image type

Table 7.11: The mean of difference scores and standard deviation for each algorithm

Algorithm	Mean	Std. Deviation	N
Backward	43.8773	6.00475	26
Forward	49.3993	7.34086	26
Novel	13.3297	8.34408	26

Table 7.12: The mean of difference scores and standard deviation for two types of images

Image Type	Mean	Std. Deviation	N
Real	36.4799	17.04714	39
Synthetic	34.5910	18.13963	39

Table 7.13: The mean of difference scores for the interaction of image type and algorithms (Image_type*Algorithms).

Image Type	Algorithms	Mean Score
Real	Backward	41.788
	Forward	51.817
	Novel	15.835
Synthetic	Backward	45.967
	Forward	46.982
	Novel	10.824

Table 7.14: The mean of difference scores for the interaction of image type and algorithms (Algorithms* Image_type).

Image Type	Algorithms	Mean Score
Backward	Real	41.788
	Synthetic	45.967
Forward	Real	51.817
	Synthetic	46.982
Novel	Real	15.835
	Synthetic	10.824

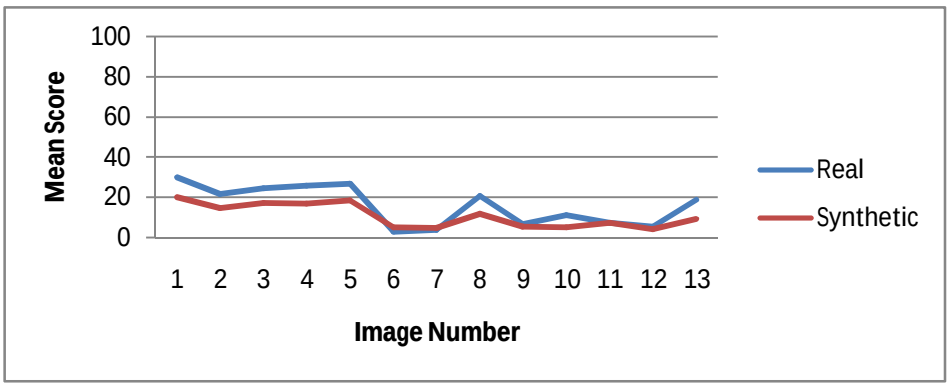


Figure 7.15: Mean of difference scores for the novel algorithm under different types of images

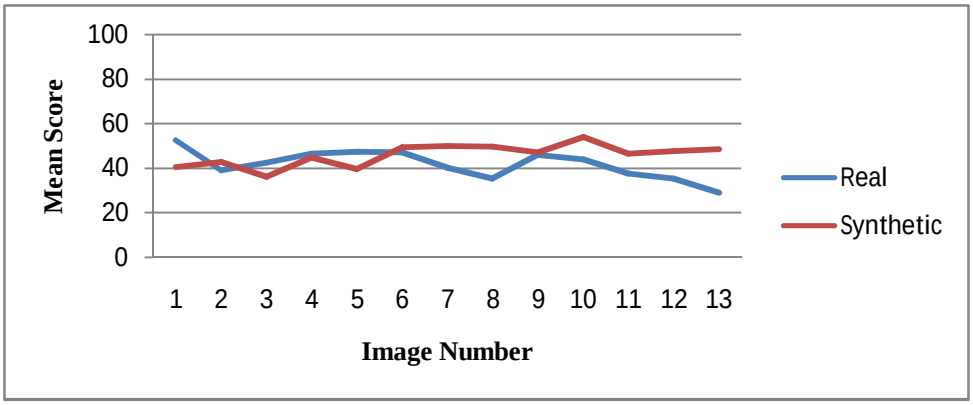


Figure 7.16: Mean of difference scores for the backward algorithm under different types of images

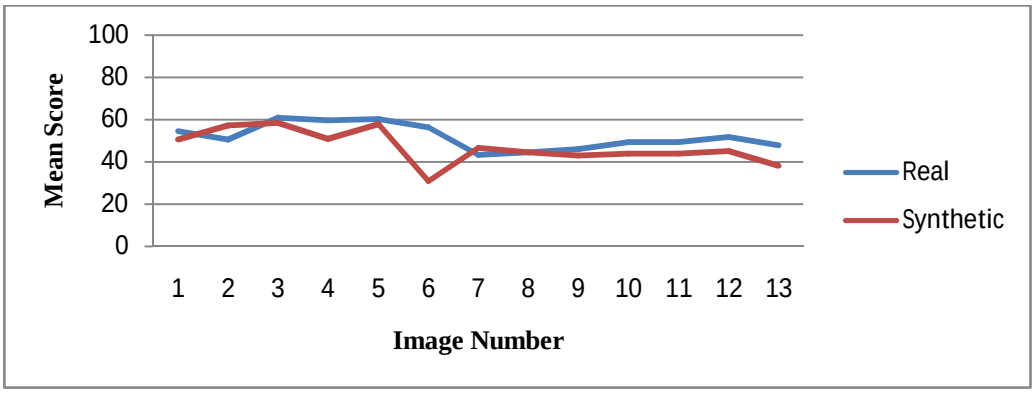


Figure 7.17: Mean of difference scores for the forward algorithm under different types of images

7.3 Conclusion

We have compared our algorithm results with two well-known interpolation algorithms (backward- and forward-disparity compensation) qualitatively using PSNR and quantitatively using human-trial test; we have shown that the proposed algorithm results outperform these other algorithms. Also, we conducted other quantitative comparisons to show the following: Firstly, the effectiveness of the proposed hole-filling technique compared to using a median filter to fill the holes and cracks that are generated in the virtual disparity; secondly, the effectiveness of linear interpolation compared to non-linear interpolation to estimate the texture of the intermediate image; thirdly, the effectiveness of the proposed inpainting technique compared to using other existing methods to inpaint the occlusion areas;

fourthly, the step-by-step quality improvement in the proposed IVR algorithm. The performance of the proposed algorithm is tested under 13 real images and 13 synthetic images. The proposed algorithm gains a higher PSNR under both types of images. Moreover, analysis of experimental results obtained from 21 participants confirmed that the reconstructed images from our proposed algorithm have very high quality comparing with the reconstructed images from the backward and forward projection algorithms.

Table 7.15: Depicts PSNR at two positions, mean of difference scores, Standard Deviation, and the minimum and maximum score from 21 participants for the real Images.

Image Type		Image Number	Image Name	PSNR		Mean Score	Std
				0.25	0.75		
Real images	Backward	1	RB1	28.42	26.00	52.62	16.30
		2	RB2	22.81	20.90	39.05	21.28
		3	RB3	26.73	22.82	42.52	23.71
		4	RB4	27.78	25.48	46.57	15.35
		5	RB5	27.25	25.79	47.52	16.61
		6	RB6	23.97	20.57	47.10	19.70
		7	RB7	28.02	26.17	40.24	17.02
		8	RB8	33.05	29.41	35.52	15.36
		9	RB9	26.18	23.02	45.95	16.23
		10	RB10	27.11	25.40	44.00	13.49
		11	RB11	31.91	29.51	37.62	17.74
		12	RB12	30.24	28.91	35.33	14.64
		13	RB13	28.66	27.20	29.19	17.65
	Forward	1	RF14	17.87	18.67	54.43	20.32
		2	RF15	20.43	19.23	50.57	19.11
		3	RF16	20.42	22.11	60.86	23.19
		4	RF17	21.31	22.92	59.57	17.62
		5	RF18	23.31	23.67	60.29	17.79
		6	RF19	20.61	20.09	56.19	22.25
		7	RF20	24.73	25.19	43.29	18.06
		8	RF21	27.19	27.14	44.43	20.61
		9	RF22	20.22	20.49	45.86	25.99
		10	RF23	22.48	23.76	49.33	18.71
		11	RF24	25.02	24.70	49.29	15.53
		12	RF25	25.86	27.26	51.86	16.52
		13	RF26	25.52	25.40	47.67	17.87
	Novel	1	RN27	37.02	38.56	29.86	14.47
		2	RN28	36.05	36.57	21.57	16.51
		3	RN29	35.38	36.03	24.62	18.66
		4	RN30	35.96	37.08	25.81	15.46
		5	RN31	36.11	36.97	26.71	15.21
		6	RN32	41.52	40.86	3.05	5.21
		7	RN33	33.98	34.51	20.86	6.66
		8	RN34	40.26	37.71	3.76	15.59
		9	RN35	39.82	40.99	6.62	8.18
		10	RN36	38.80	40.30	11.10	12.14
		11	RN37	38.26	37.60	7.52	8.87
		12	RN38	33.63	36.77	5.43	6.82
		13	RN39	31.77	34.00	18.95	15.74

Table 7.16: Depicts PSNR at two positions, mean of difference scores, Standard Deviation, and the minimum and maximum score from 21 participants for the synthetic Images.

Image Type		Image Number	Image Name	PSNR		Mean Score	Std
				0.25	0.75		
Synthetic Images	Backward	1	SB1	30.42	32.75	49.57	14.23
		2	SB2	24.26	25.07	50.14	19.10
		3	SB3	27.09	28.30	49.67	17.89
		4	SB4	26.13	27.35	47.14	17.54
		5	SB5	27.99	28.72	54.14	15.97
		6	SB6	25.67	26.95	46.67	21.50
		7	SB7	22.36	22.73	47.62	22.04
		8	SB8	24.96	24.22	48.48	18.34
		9	SB9	29.94	28.06	40.43	17.70
		10	SB10	28.30	25.84	42.81	14.00
		11	SB11	29.01	26.73	36.29	20.52
		12	SB12	29.01	27.31	44.95	17.41
		13	SB13	27.91	25.63	39.67	16.23
	Forward	1	SF14	40.17	36.06	30.86	15.48
		2	SF15	27.06	31.17	46.71	21.38
		3	SF16	29.56	31.21	44.57	23.12
		4	SF17	29.46	32.72	43.00	23.24
		5	SF18	30.92	36.16	43.81	20.09
		6	SF19	29.76	34.41	43.86	19.62
		7	SF20	24.00	26.07	45.00	14.28
		8	SF21	24.03	27.22	38.14	20.83
		9	SF22	24.27	24.27	50.62	18.50
		10	SF23	22.23	22.28	57.24	20.40
		11	SF24	21.70	22.05	58.33	17.21
		12	SF25	23.67	23.84	50.90	19.72
		13	SF26	22.25	22.28	57.71	20.25
	Novel	1	SN27	49.79	49.79	5.29	8.01
		2	SN28	38.29	33.50	4.90	8.48
		3	SN29	39.90	39.93	11.76	11.06
		4	SN30	37.74	37.76	5.52	12.68
		5	SN31	47.28	43.34	5.05	6.63
		6	SN32	35.89	38.05	7.38	10.00
		7	SN33	43.24	43.24	4.10	7.85
		8	SN34	43.26	43.26	9.33	8.97
		9	SN35	39.67	40.62	20.10	13.85
		10	SN36	40.03	41.04	14.81	12.34
		11	SN37	40.23	40.11	17.19	13.85
		12	SN38	40.62	41.52	16.90	14.19
		13	SN39	38.60	39.22	18.38	14.67

CHAPTER

8

Conclusions and Future Work

This chapter offers a general discussion on the main contribution of this research, followed by a list of open directions for future research issues.

8.1 Conclusions

The main goal of the research presented in this thesis is to generate additional images from the available reference images. Intermediate view reconstruction (IVR) is considered an effective solution for the generation process instead of using a multi-camera system. Specifically, this approach, which is considered the main goal of this thesis, produces continuous viewpoints from the available stereo image pair to provide a natural look-around feeling with comfortable viewing. IVR can be used to generate the content of the multiscopic 3D displays that have emerged recently, and to generate the different viewpoints to Free-viewpoint TV (FTV). Actually, the content of the multiscopic 3D displays are rarely available and their generation is considered a hard task since they need high-quality images. Although the IVR is considered a good approach to find additional images, there are problems with the generation process and they are considered difficult challenges for researchers to solve. Therefore, in this thesis we explored intermediate view challenges and subsequently proposed several novel algorithms to handle these challenges; these novel algorithms were then employed in a highly robust intermediate view reconstruction algorithm. In addition, we focused on generating high-quality images since these are a requirement of 3D displays where these images are tested.

8.1.1 Thesis contribution

The contribution of this research is the development of a new algorithm to find intermediate images with high quality. Detection and handling occlusion problems are considered the main challenges of intermediate view reconstruction [17]. In addition, a number of image artifices are generated through formation of the intermediate view and these must be resolved; they include generation of holes and cracks due to image resampling, generation of ghost contours due to the disparity's sharp edges, and appearance of disoccluded area due to cameras changing position. Moreover, because the intermediate image is generated from interpolation of more than one view, it is necessary to preserve the discontinuities in the reconstructed image, and this is considered a challenge for the reconstruction process.

Therefore, several novel algorithms have been specifically designed to overcome these challenges, and they are employed in a highly robust intermediate view reconstruction. So, this thesis offers the following main contributions:

1. Developing an efficient novel occlusion detection algorithm to detect the occluded/newly-exposed area based on the physical movements of the objects between the stereo images. Our algorithm has many advantages; for example, the occlusion area is detected without any fragmentation, the results are consistent under different types of images and matching algorithms, it needs few input parameters, and it is easy to implement. Qualitative and quantitative measures confirmed that the proposed algorithm detects the occlusion areas with high accuracy and a lower error rate.
2. Developing an accurate block-based disparity inpainting algorithm for occlusion area recovery. This novel technique is specialized for occlusion areas, filling in disparity images. Our algorithm has many advantages; for example, the occluded area is filled without any blurriness, edge discontinuities are preserved, and the line curvature is continued. We performed different analyses and computer simulations which confirmed that the proposed technique inpaints the occlusion area with high accuracy and a lower error rate.

3. These proposed algorithms are employed in a novel occlusion-aware intermediate view reconstruction. This novel algorithm focuses on how to find an intermediate image with a given disparity between two input images. An improvement on the backward-projection of the intermediate view is proposed by adding occlusion awareness to the reconstruction algorithm and proposing three quality improvement techniques to reduce image artifices, as follows:

   3.1. Filling the re-sampling holes: instead of using a median filter to fill cracks and holes produced by image re-sampling, we proposed an object-based filling technique that fills the hole located in a foreground object or around its edges by the object's intensity. Subjective and objective evaluations proved that the proposed technique produces more accurate results around object boundaries

   3.2. Removing ghost contours from the virtual view and restoring them to their correct place by extending the newly-exposed area three pixels in the direction where the ghost contour is located.

   3.3. Handling the disocclusion areas: the disocclusion areas are recovered from one of the inpainted disparities upon the calculated visibility map.

   After that, the texture of the intermediate image is reconstructed either from left or right images based on a robust and efficient selective strategy. This selection is performed upon the visibility of each image point in the intermediate image, which is determined by a novel approach. Specifically, this approach is developed to detect the occluded/ newly-exposed area. Depending on this information, the visibility of the image points in the intermediate image will be determined.

4. Conducting a subjective human trial to assess the quality of the reconstructed intermediate image on 3D display relative to the reference intermediate image based on the ITU-R 500 recommendation and comparing it to the reconstructed images by backward and forward algorithms.

8.2 Results and discussion on the individual chapters

- ***Chapter 4: Newly-Exposed/Occlusion Detection (NEOD) Algorithm***

  In this chapter, a novel occlusion detection algorithm has been proposed to overcome the deficiencies of the previous occlusion detection algorithms. The performance of the proposed algorithm is tested under 2 synthetic datasets and 3 real datasets. We have compared our algorithm results with three well-known occlusion detection algorithms (LRC, ORD, and OCC) qualitatively and quantitatively, and have shown that the proposed algorithm results outperform these algorithms under both types of dataset. We performed different analyses and computer simulations which confirmed that the proposed algorithm detects the occlusion areas with a high accuracy (0.9939 and .9823) and low error rate (0.0061 and 0.0177) for the well-known datasets Tsukuba and Teddy. This computer simulation also confirmed that the proposed algorithm detects the occlusion for two synthetic datasets with high accuracy (1, 0.9967) and low error rate (0, 0.0033).

- ***Chapter 5: Block-based Inpainting Technique for Occlusion Area (BITO)***

  In this chapter, a novel inpainting technique has been proposed to overcome the deficiencies of the previous inpainting techniques. We have compared our algorithm results with four well-known inpainting algorithms (HEDI, Exemplar-based, FOE, and ROIFILL) qualitatively and quantitatively using PSNR; we have shown that the proposed algorithm results outperform these algorithms. HEDI is specialized for occlusion filling and the other three are for general image inpainting, specifically for object removal or image restoration. Our algorithm has many advantages; for example, the occluded area is filled without any blurriness since it does not depend on any smoothness diffusion. By exploiting the edge map of the underlying stereo image, edge discontinuities are diffused in the occlusion area in the disparity map, and the line curvature is continued. We performed different analyses and computer simulations which confirmed that the proposed technique inpaints the occlusion area with high accuracy. This is shown using PSNR calculation, where the proposed BITO algorithm gained on average a higher PSNR of ~0.16, ~6, ~12, and ~17 db than

the Exemplar-based, HEDI, FOE and ROIFILL inpainting techniques, respectively

- ***Chapter 6: Occlusion-Aware Intermediate View Reconstruction***

  In this chapter, a novel intermediate view reconstruction algorithm has been proposed to overcome the deficiencies of the previous reconstruction algorithms. This novel algorithm focuses on how to find an intermediate image with a given disparity between two input images. As the detection and handling occlusion problems are considered among the challenges of the intermediate view reconstruction, we employed the algorithms proposed in chapters 4 and 5 in a novel occlusion-aware intermediate view reconstruction. We proposed three quality improvement techniques to reduce image artifices. First, instead of using a median filter to fill cracks and holes produced by image re-sampling, we proposed an object-based filling technique that fills the hole located in a foreground object or around its edges by the object's intensity. Subjective and objective evaluations proved that the proposed technique produces more accurate results around object boundaries. Second, we removed the ghost contours from the virtual view and restored them to their correct place by extending the newly-exposed area three pixels in the direction where the ghost contour is located. Third, since the novel reconstruction algorithm detects and inpaints the occlusion areas in the two input disparity maps, the disocclusion areas are recovered from one of the input images. Inpainting the double-occlusion area as a post-processing step is considered a reasonable solution to this problem since it is not visible in both reference images, so inpainting in the disparity maps is not an applicable solution.

- ***Chapter 7: Occlusion-Aware Intermediate View Reconstruction Results***

  We have compared our algorithm results with two well-known interpolation algorithms (backward- and forward-disparity compensation) qualitatively using PSNR and quantitatively using a human-trial test; we have shown that the proposed algorithm results outperform these algorithms. Also, we conducted other quantitative comparisons to show the following: Firstly, the effectiveness of the proposed hole-filling technique compared to using a median filter to fill

the holes and cracks that are generated in the virtual disparity; secondly, the effectiveness of linear interpolation compared to non-linear interpolation to estimate the texture of the intermediate image; thirdly, the effectiveness of the proposed inpainting technique compared to using other existing methods to inpaint the occlusion areas; fourthly, the step-by-step quality improvement in the proposed IVR algorithm.

The performance of the proposed algorithm is tested under 13 real images and 13 synthetic images. The proposed algorithm gains a higher PSNR under both types of images. Moreover, analysis of experimental results obtained from 21 participants confirmed that the reconstructed images from our proposed algorithm have very high quality comparing with the reconstructed images from the backward and forward projection algorithms.

8.3 Future Works

In regard to future work, we recommend the following improvement issues.

8.3.1 FTV Application

FTV has been developed to provide the viewer with different viewpoints as he/she moves. This is achieved by displaying multiple views and the viewer selecting the preferred viewpoint by changing his/her position. IVR is considered a good approach for generating the content of FTV. Since the proposed IVR algorithm generates high-quality images on a 3D display, we expect that those generated images will be good enough for use on an FTV. Therefore, we recommend testing our reconstructed images from the proposed reconstruction algorithm on an FTV as a future project.

8.3.2 Assumption Validation

In this thesis, we determined the relationship between the image points in the left and right image using the Constant Image Brightness (CIB) assumption. This assumption assumes that the corresponding image points between the stereo images have the same luminance value. In fact, CIB assumption is not always true since an object has reflecting surfaces and this reflection value might be changed when viewed from a different position. Due to the difficulty of this situation, we made this assumption in

this thesis. Further research would be desirable to take into consideration the surface reflectance problem in the stereo matching problem.

8.3.3 Improving Forward-projection approach

The reconstructed image by forward algorithm suffers from many corrupted areas such as generation of many holes and cracks, and corruption at the occlusion areas. The holes and cracks are generated in the reconstructed image because some points in the intermediate image are not assigned intensity values (undefined points); a texture synthesis can be used to cover these cracks but the image will be distorted. Unawareness of this approach for the occlusion areas causes a corruption in such areas.

The proposed improvement techniques for the backward-projection approach can be exploited to improve the forward-projection approach. This could be done by adding occlusion awareness to the reconstruction algorithm, and employing the proposed object-based hole filling technique to fill the generated holes and cracks in the final reconstructed image. Although this technique fills the holes and cracks generated in the disparity map, it can be exploited by labelling these holes and cracks in the disparity map, and then inversely reconstructing them based on their positions.

8.3.4 Implementation Issues

In this thesis, we used MATLAB as a software environment to implement the proposed algorithms due to the availability of some built-in functions. Because some applications need to generate multiple images in real time and the MATLAB environment by its nature is slow, implementing the proposed algorithm in C will speed up the generation process. The OpenCV library can be used, as this is considered a suitable software environment to manipulate the images. Inpainting the occlusion areas is the most time-consuming step even though it is faster than the other accurate algorithms; re-implementing this step in C might speed up the generation process.

Bibliography

1. N.J. Wade, H. Ono, and L. Lillakas, *Historical Perspective: Leonardo da Vinci's Struggles with Representations of Reality*. Leonardo, the International Society for the Arts, Sciences and Technology (ISAST), 2001. **34**: p. 231–235.

2. J. Beck, *Leonardo's rules of painting*. 1979, Oxford: Phaidon Press.

3. C. Wheatstone, *Contributions to the Physiology of Vision.—Part the First: On some remarkable, and hitherto unobserved, Phenomena of Binocular Vision.* Philosophical Transactions of the Royal Society June 1838. **128**: p. 371 - 394.

4. L. Spiro, *A brief history of the stereographs and stereoscopes.* Oct.30, 2006, Connexions, <http://cnx.org/content/m13784/latest/>.

5. N.S.Holliman, *Three-Dimensional Display Systems*, in *Handbook of Optoelectronics*, Ed.J.P.Dakin and R.G.W.Brown, Editors. May, 2006, Taylor and Francis.

6. W.R. Watkins, G.D. Heath, M.D. Phillips, J.M. Valeton, and A. Toet, *Search and target acquisition: single line of sight versus wide baseline stereo.* Optical Engineering, 2001. **40**: p. 1914–1927.

7. G. Moraglia and B. Schneider, *On binocular unmasking of signals in noise: further tests of the summation hypothesis.* Vision Research, 1992. **32**: p. 375–385.

8. W.A. Ijsselsteijn, H.D. Ridder, and J. Vliegen, *Subjective evaluation of stereoscopic images: effects of camera parameters and display duration.* IEEE Transactions on Circuits and Systems for Video Technology 2000. **10**: p. 225–233.

9. N.A. Valyus, *Stereoscopic cinematography and television* in *Stereoscopy*. 1966, Focal Press: New York.

10. T. Okoshi, *Three-Dimensional Imaging Techniques*. 1976: Academic Press.

11. B. Javidi and F. Okano, *Three-Dimensional Television, Video, and Display Technologies*. 2002: Springer-Verlag.

12. W. Matusik and H. Pfister, *3D TV: A Scalable System for Real-Time Acquistion, Transmission and Autostereoscopic Display of Dynamic Scenes.* ACM Transactions on Graphics (TOG) SIGGRAPH, 2004. **23**: p. 814-824.

13. S. Ma and P. Milgram. *Investigation of potential benefits of stereoscopic video for visual detection in turbid underwater environments.* in *Proceedings of SPIE*. 2001.

14. N.A. Dodgson. *Variation and extrema of human interpupillary distance*. in *Proceedings of the SPIE* 2004. California, USA.

15. S. Pastoor, *3D-television: A survey of recent research results on subjective requirements.* Signal Processing: Image Communication, 1991. **4**: p. 21-32.

16. M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo, *Free-viewpoint TV, A review of the ultimate 3DTV and its related technologies.* IEEE Signal Processing Magazine, 2011. **28**: p. 67-76.

17. S. Ince and J. Kornad, *Occlusion-aware view interpolation.* EURASIP Journal on Image and Video Processing 2008.

18. E.R. Davies, *Machine Vision: Theory, Algorithms, Practicalities.* 3rd ed. 2005: Morgan Kaufmann, Elsevier.

19. K.N. Ogle, *Researches in Binocular Vision*. 1964: Hafner Publishing Co. Ltd,.

20. E.B. Goldstein, *Sensation and Perception*. 6th ed. 2002: Wadsworth.

21. S. Pastoor. *Human Factors of 3d Imaging: Results of Research at Heinrich-Hertz-Institute, Berlin* in *Proc. International Display Workshops '95*. 1995.

22. S.M. Faris. *Novel 3D-stereoscopic imaging technology*. in *Proceedings of the SPIE*. 1994.

23. N.A. Dodgson, *Autostereoscopic 3D displays.* IEEE Computer Society, 2005. **38**: p. 31-36.

24. D.Ezra, G.Woodgate, J.Harrold, B.Omar, N.Holliman, and L. Shaprio. *New Auto-stereoscopic display system.* in *Proceedings of the SPIE*. 1995.

25. D. Ezra, G.J. Woodgate, and B. Omar, *Autostereoscopic Directional Display Apparatus,* in *US Patent 5726800*. 1998.

26. F.E. Ives, *Parallax stereogram and process for making same,* in *U.S. Patent 725567*. 1903.

27. G. Woodgate, J. Harrold, M. Jacobs, R. Moseley, and D. Ezra. *Flat panel autostereoscopic displays - characterisation and enhancement.* in *Proceedings of the SPIE*. 2000.

28. D.J. Montgomery, G.J. Woodgate, and D. Ezra, *Parallax barrier, autostereoscopic 3D picture and autostereoscopic 3D display. Patent No. GB 2352573*. 2001.

29. G.J. Woodgate and J. Harrold, *High Efficiency Reconfigurable 2D/3D Autostereoscopic Display*. SID Symposium Digest of Technical Papers, 2003. **34**: p. 394-397.

30. G. Woodgate, D. Ezra, J. Harrold, N. Holliman, G. Jones, and R. Moseley. *Observer tracking autostereoscopic 3D display systems.* in *Proceedings of the SPIE*. 1997.

31. P. Harmann. *Retroreflective Screens and Their Application to Autostereoscopic Displays.* in *Proceedings of the SPIE*. 1997.

32. C. Vanberkel and J.A. Clarke. *Characterisation and Optimisation of 3D-LCD Module Design* in *Proceedings of the SPIE*. 1997.

33. D.F. Winnek, *Composite stereography*, in *U.S. Patent 3409351*. 1968.

34. G. Lippmann, *Epreuves reversibles donnant la sensation du relief*. . Journal of Physics, 1908: p. 821–825.

35. S. Nakajima, K. Nakamura, K. Masamune, I. Sakuma, and T. Dohi, *Three-dimensional medical imaging display with computer-generated integral photography*. Computerized Medical Imaging and Graphics 2001. **25**: p. 235-241.

36. O. Fumio, H. Haruo, A. Jun, Y. Mitsuho, and Y. Ichiro, *Three-Dimensional Television System Based on integral Photography*, in *Three-Dimensional Television, Video, and Display Technologies*, B. JAVIDI and F. OKANO, Editors. 2002, Springer-Verlag.

37. H.E. Ives, *The projection of parallax panoramagrams.* Journal of the Optical Society of America, 1931. **21**: p. 397–409.

38. N.A. Dodgson, J.R. Moore, S.R. Lang, G. Martin, and P. Canepa, *Time-Sequential Multiprojector Autostereoscopic 3D Display*. Journal of the Society for Information Display, 2000. **8**: p. 69-176.

39. N.A. Dodgson, J.R. Moore, S.R. Lang, G. Martin, and P. Canepa. *A 50-Inch Time-Multiplexed Autostereoscopic Display*. in *Proceedings of SPIE*. 2001.

40. J.D. Montes and P. Campoy. *A New Three-Dimensional Visualisation System Based on Angular Image Differentiation*. in *Proceedings of SPIE*. 1995.

41. K. Matsumoto and T. Honda. *Research of 3-D Display Using Anamorphic Optics*. in *Proceedings of SPIE*. 1997.

42. R. Hartely and A. Zisserman, *Multiple View Geometry in Computer Vision.* 2nd ed. 2003: Cambridge University Press.

43. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* 2006, Cambridge, UK: Cambridge University Press.

44. R.I. Hartley, *Theory and practice of projective rectification.* International Journal of Computer Vision, 1999. **35**: p. 115-127.

45. J. Zhou and B. Li, *Image rectification fore stereoscopic visualization.* Optical Society of America, 2008. **25**.

46. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with OpenCV Library*. 1st ed. Septemper 2008: O'Reilly Media, Inc.

47. J.G. Fryer and D.C. Brown, *Lens distortion for close-range photogrammerty.* Photogrammetric Engineering and Remote Sensing 1986. **52**: p. 51-58.

48. B. Horn and B. Schunck, *Determining optical flow.* Artificial Intelligence, 1981. **17**: p. 185-203.

49. A. Redert, E. Hendriks, and J. Biemond, *Correspondence estimation in image pairs.* IEEE Signal Processing Magazine, 1999. **16**: p. 29–46.

50. M. Accame, F.G.B.D. Natale, and D. Giusto. *Hierarchical block matching for disparity estimation in stereo sequences.* in *Proceedings of International Conference on Image Processing*. 1995.

51. L. Zhang. *Hierarchical block-based disparity estimation using mean absolute difference and dynamic programming*. in *Proceedings of International Workshop on Very Low Bitrate Video Coding*. 2001.

52. T. Kanade and M. Okutomi, *A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994. **16**: p. 920-932.

53. J. Konrad and Z.-D. Lan. *Dense disparity estimation from feature correspondences.* in *Proceedings of the SPIE*. 2000.

54. S.Sharghi and F.A. Kamangar. *Geometric feature-based matching in stereo images*. in *Proceedings of Information Decision and Control*. 1999.

55. J. Konrad and Z.-D. Lan. *Dense-disparity estimation from feature correspondences*. in *Proceedings of SPIE*. 2000.

56. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High accuracy optical flow estimation based on a theory for warping*. Proceedings of the 8th European Conference on Computer Vision, 2004. **4**: p. 25-36.

57. R. March, *Computation of stereo disparity using regularization*. Pattern Recognition Letters, 1988. **8**: p. 181–187.

58. H. Kim and K. Sohn, *3D reconstruction from stereo images for interactions between real and virtual objects*. Signal Processing: Image Communication, 2005. **20**(1): p. 61-75.

59. A. Maki, L. Bretzner, and J.-O. Eklundh. *Local Fourier phase and disparity estimates: An analytical study*. in *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns*. 1995.

60. U. Ahlvers, U. Zoelzer, and S. Rechmeier. *FFT-based disparity estimation for stereo image coding*. in *IEEE International Conference on Image Processing*. 2003.

61. A. Cozzi, B. Crespi, F. Valentinotti, and F. Wrgtter, *Performance of phasebased algorithms for disparity estimation*. Machine Vision and Applications, 1997. **9**: p. 334–340.

62. M. Chang, M. Sezan, and A. Tekalp. *An algorithm for simultaneous motion estimation and scene segmentation*. in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. 1994.

63. J. Konrad and E. Dubois, *Bayesian estimation of motion vector fields*. EEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(9): p. 910–927.

64. O. Veksler, *Stereo correspondence by dynamic programming on a tree.* Computer Society Conference on Computer Vision and Pattern Recognition, 2005. **2**: p. 384- 390.

65. V. Kolmogorov and R. Zabih. *Computing visual correspondence with occlusions using graph cuts.* in *Proceedings of the Eighth IEEE International Conference on Computer Vision.* 2001.

66. J. Sun, N.-N. Zheng, and H.-Y. Shum, *Stereo Matching Using Belief Propagation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003. **25**(7): p. 787 - 800

67. L. Alvarez, R. Deriche, J. Snchez, and J.Weickert, *Dense disparity map estimation respecting image discontinuities : A PDE and scale-space based approach.* Journal of Visual Communication and Image Representation, 2002. **13**: p. 3–21.

68. L. Alvarez, J. Weickert, and J. Sanchez, *Reliable estimation of dense optical flow fields with large displacements.* International Journal of Computer Vision, 2000. **39**: p. 41–56.

69. G. Sapiro, *Geometric Partial Differential Equations and Image Analysis.* 2001: Cambridge University Press.

70. T. Chan, J. Shen, and L. Vese, *Variational PDE models in image processing.* Notices of the American Mathematical Society, 2003. **50**(1): p. 14-26.

71. S. Ince, J. Kornad, and C. Vazquez. *Spline-based intermediate view reconstruction*. in *Proceedings of SPIE*. 2007.

72. L. Do, S. Zinger, and P.H.N.D. With. *Quality improving techniques for free-viewpoint DIBR*. in *Proceedings of SPIE: Stereoscopic Dispalys and Virtual Reality Systems*. 2010.

73. A. Mancini and J. Konrad. *Robust quadtree-based disparity estimation for the reconstruction of intermediate stereoscopic images.* in *proceedings of the SPIE*. 1998.

74. R. Braspenning and M.O.D. Beeck. *Efficient view synthesis from uncalibrated Stereo*. in *Proceedings of the SPIE* 2006.

75. A.R. Mansouri and J. Konrad, *Bayesian winner-take-all reconstruction of intermediate views from stereoscopic images.* IEEE Transactions on Image Processing, 2000. **9**: p. 1710-1722.

76. J. Mcveigh, M. Siegel, and A. Jordan, *Intermediate view synthesis considering occluded and ambiguously referenced image regions.* Signal Process: Image Communication, 1996. **9**: p. 21-28.

77. D. Scharstein. *Stereo vision for view synthesis.* in *Proceedings of IEEE Conference Computer Vision Pattern Recognition.* 1996.

78. R. Depommier and E. Dubois. *Motion estimation with detection of occlusion areas.* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing.* 1992.

79. J. Weng, N. Ahuja, and T.S. Huang. *Two-view matching*. in *Proc. IEEE International Conference on Computer Vision* 1988.

80. A.Luo and H. Burkhardt, *An intensity-based cooperative bidirectional stereo matching with simultaneous detection of discontinuities and occlusions.* International Journal of Computer vision, 1995. **vol. 15**: p. 171-188.

81. R. Trapp, S. Drue, and G. Hartmann. *Stereo Matching with Implicit Detection of Occlusions.* in *Proceedings of the 5th European Conference on Computer Vision*. 1998.

82. A.L. Yuille and T. Poggio, *A generalized ordering constraint for stereo correspondence*, in *MIT Computer Science and Artificial Intelligence Lab 777*. 1984: Cambridge.

83. D. Geiger, B. Ladendorf, and A. Yuille, *Occlusion and binocular stereo.* International Journal of Computer Vision, 1995. **vol. 14**(s): p. 211-226.

84. S.S. Intille and A.F. Bobick. *Disparity-space image and large occlusion stereo.* in *Proceedings of the European Conference on Computer Vision.* 1994.

85. D. Marr and T. Poggio, *Cooperative computation of stereo disparity.* Science, 1976. **194**: p. 283-287.

86. S.B. Pollard, J.E. Mayhew, and J.P. Frisby, *PMF: A stereo correspondence algorithm using a disparity gradient limit.* Perception, 1985. **14**(4): p. 449-470.

87. J. Sun, Y. Li, S.B. Kang, and H.-Y. Shum. *Symmetric stereo matching for occlusion handling*. in *Proceedings of IEEE Conference Computer Vision Pattern Recognition*. 2005.

88. S. Ince and J. Konrad. *Geometry-based estimation of occlusions from video frame pairs.* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. 2005.

89. F. Boughorbel. *Adaptive filter for depth from stereo and occlusion detection.* in *In stereoscopic Displays and Applications XIX, Proceeding of SPIE*. 2008.

90. G. Egnal and R.P. Wildes, *Detecting binocular half-occlusions: empirical comparisons of five approaches.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **vol. 24**: p. 1127-1133.

91. J. Park and S. Inoue. *Arbitrary view generation using multiple cameras.* in *Proceedings of IEEE International Conference on Image Processing*. 1997.

92. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. *Image inpainting.* in *In SIGGRAPH'00: Proceedings of the 27th annual conference on computer graphics and interactive techniques.* 2000.

93. S. Ince and J. Konrad, *Occlusion-aware optical flow estimation.* IEEE Transactions on Image Processing, 2008. **17**.

94. A. Criminisi, P. Perez, and K. Toyama. *Object Removal by Exemplar-Based Inpainting*. in *IEEE Transactions on Image Processing*. 2004.

95. M. Bertalmio, L. Vese, G. Sapiro, and S. Osher., *Simultaneous structure and texture image inpainting.* IEEE Transaction on Image Processing, 2003. **12**: p. 882-889.

96. P. Harrison. *A non-hierarchical procedure for re-synthesis of complex texture*. in *Proceedings of International Conference in Central Europe on Computer Graphics, Visualazation and Computer Vision*. 2001.

97. L. Azzari, F. Battisti, A. Gotchev, M. Carli, and K. Egiazarian, *A Modified Non-local Mean Inpainting Technique for Occlusion Filling in Depth-Image Based Rendering.* Proceedings of SPIE: Stereoscopic Displays and Applications XXII, 2011.

98. A. Colombari, A. Fusiello, and V. Murino. *Continuous parallax adjustment for 3D-TV*. in *Proceedings of IEE 2nd European Conference on Visual Media Production*. 2005.

99. S. Roth and M.J. Black, *Fields of Experts.* International Journal of Computer Vision, 2009. **82**: p. 205–229.

100. L. Zhang, W.J. Tam, and D. Wang. *Stereoscopic image generation based on depth images*. in *Proceeding of IEEE International Conference on Image Processing (ICIP 04)*. 2004.

101. L. Zhang and W.J. Tam, *Stereoscopic image generation based on depth images for 3D TV*. IEEE Transactions on Broadcasting, 2005. **51**(2): p. 191-199.

102. K.-J. Oh, S. Yea, and Y.-S. Ho. *Hole-Filling Method Using Depth Based In-Painting For View Synthesis in Free Viewpoint Television (FTV) and 3D*

Video. in *Proceedings of the 27th conference on Picture Coding Symposium.* 2009.

103. C. V´Azquez, W.J. Tam, and F. Speranza. *Stereoscopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering*. in *Proceeding of SPIE: 3D TV, video, and display V*. 2006.

104. L. Azzari, F. Battisti, and A. Gotchev. *Comparative Analysis of Occlusion-Filling Techniques in Depth Image-Based Rendering for 3D Videos.* in *Proceedings of ACM: the 3rd Workshop on Mobile Video Delivery (MoViD).* 2010.

105. S.B. Kang, *A survey of image-based rendering techniques.* 1997, Cambridge Research Laboratory of Digital Equipment Corporation.

106. H. Shum, S. Kang, and S. Chan, *Survey of image-based representations and compression techniques.* IEEE Transactions on Circuits Systems and Video Technology, 2003. **13**(11): p. 1020-1037.

107. M. Levoy and P. Hanrahan. *Light field rendering*. in *SIGGRAPH'96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques.* 1996.

108. S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. *The lumigraph.* in *SIGGRAPH'96: Proceedings of the 23rd annual conference on computer graphics and interactive techniques.* 1996.

109. P.E. Debevec, G. Borshukov, and Y. Yu, *Efficient view-dependent image-based rendering with projective texture-mapping.* In 9th Eurographics Rendering Workshop, 1998.

110. W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. Mcmillan. *Image-based visual hulls.* in *SIGGRAPH'00: Proceedings of the 27th annual conference on computer graphics and interactive techniques.* 2001.

111. M. Gotfryd, M. Wegner, and M. Domanski, *View synthesis software and assessment of its performance video,* in *ISO/IEC JTC1/SC29/WG11 MPEG/M15672*. 2008: Germany.

112. Y. Jeong, Y. Kwan, Y. Han, Y. Jung, and D. Park, *Depth-image-based rendering (DIBR) using disocclusion area restoration.* SID Symposium Digest of Technical Papers, 2009: p. 119-122.

113. Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto. *View generation with 3D warping using depth information for FTV*. in *IEEE 3DTV conference: The True Vision-Capture, Transmission and Dispaly of 3D Video*. 2008. Turkey.

114. A. Telea, *An image inpainting technique based on the fast marching method.* Journal of Graphics Tools, 2004. **9**: p. 23-34.

115. L. Do, S. Zinger, Y. Morvan, and P.H.N.D. With, *Quality improving techniques in DIBR for free-viewpoint video,* in *IEEE 3DTV conference: The true vision-capture, transmission and display of 3D video*. 2009.

116. L. Do, S. Zinger, and P.D. With. *Objective quality analysis for free-viewpoint DIBR*. in *International Conference on Image Processing (ICIP)*. 2010.

117. L. Do, S. Zinger, and P.D. With. *Warping error analysis and reduction for depth image based rendering in 3DTV*. in *Proceedings of the SPIE*. 2011.

118. F. Devernay, S. Duchene, and A. Ramos-Peon. *Adapting stereoscopic movies to the viewing conditions using depth-preserving and artifact-free novel view synthesis*. in *Proceedings of the SPIE*. 2011.

119. P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990. **12**: p. 629–639.

120. D. Scharstein. *Matching images by comparing their gradient fields.* in *Proceedings of 12th International Conference on Pattern Recognition*. 1994.

121. A. Redert, E. Hendriks, and J. Biemond. *Synthesis of multi viewpoint images at non-intermediate positions.* in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing.* 1997.

122. J. Park and S. Inoue, *Acquisition of sharp depth map from multiple cameras.* Signal Processing: Image Communication, 1998. **14**: p. 7–19.

123. J.M. Harris. *Monocular zones in stereoscopic scenes: a useful source of information for human binocular vision?* in *Stereoscopic Displays and Application XXI, Proceeding of the SPIE*. 2010. USA.

124. J.M. Harris and L.M. Wilcox, *The role of monocularly visible regions in the perception of three-dimensional scenes.* Vision Research, 2009. **49**: p. 2666-2685.

125. A. Spoerri and S.Ullman. *The early detection of motion boundaries.* in *Proc. IEEE International Conference on Computer Vision.* 1987. London, UK.

126. B. Anderson and K. Nakayama, *Toward a general theory of stereopsis: binocular matching, occluding contours, and fusion.* Psychological Review, 1994. **Vol. 101**: p. 414-445.

127. Z. Wang and Z. Zheng, *A region based stereo matching algorithm using cooperative optimization,* in *IEEE on Computer Vision and Pattern Recognition*. 2008.

128. M.M. Oliveira, B. Bowen, and R. Mckenna. *Fast Digital Image Inpainting.* in *Proceeding of International Conference on Visualization, Imaging and Image Processing (VIIP2001)* September 2001.

129. J. Zhou and A. Robles-Kelly, *Image inpainting based on local optimization,* in *International Conference on Patteren Recongnition (ICPR)* 2010.

130. Y.K. Park, K. Jung, Y. Oh, S. Lee, J.K. Kim, G. Lee, H. Lee, K. Yun, N. Hur, and J. Kim., *Depth-image-based rendering for 3DTV service over T-DMB.* Signal Processing: Image Communication, 2009. **24** p. 122–136.

131. D. Tschumperle and R. Deriche, *Vector-valued image regularization with PDEs : A common framework for different applications.* IEEE Transaction on Pattarn Analysis and Machine Intelligence (TPAMI), 2005. **27**: p. 506–517.

132. C. Fehn. *Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV* in *Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI*. 2004.

133. S. Bhat, M. Davey, and J.G. On, *Object Removal by Exemplar-Based Inpainting*. 2005.

134. S. Roth, *Fields of Experts*. 2005, Brown University.

135. J. Zhai, K. Yu, J. Li, and S. Li. *A low complexity motion compensated frame interpolation method.* in *In Proceedings of IEEE International Symposium on Circuits and Systems*. 2005.

136. A. Mancini and J. Konrad. *Robust quadtree-based disparity estimation for the reconstruction of intermediate stereoscopic images.* in *Conference on Stereoscopic Displays and Virtual Reality Systems V*. 1998. San Jose, Ca.

137. A.R. Mansouri and J. Konrad, *Bayesian winner-take-all reconstruction of intermediate views from stereoscopic images.* Image Processing, IEEE Transactions on, 2000. **9**(10): p. 1710-1722.

138. A. Criminisi, P. Perez, and K. Toyama. *Object Removal by Exemplar-Based Inpainting*. in *Proc. of IEEE conference on computer Vision and Patteren Recognition* 2003.

139. *Methodology for the subjective assessment of the quality of television pictures,* in *Technical Report:Recommendation ITU-R BT.500-11*. 1998, International Telecommunications Union.

Appendix A

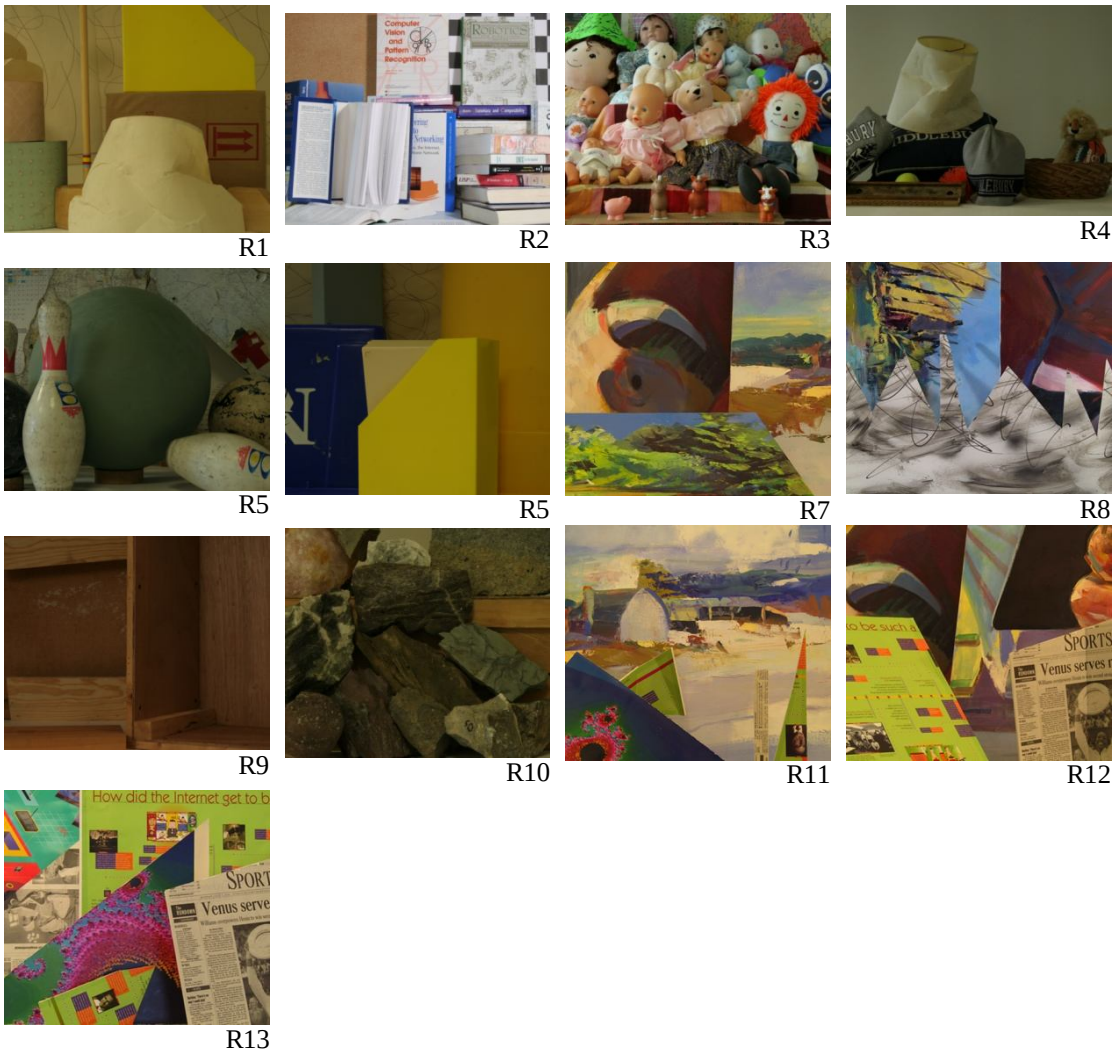


Figure A.1: Showing the 13 real images that used in the experiment

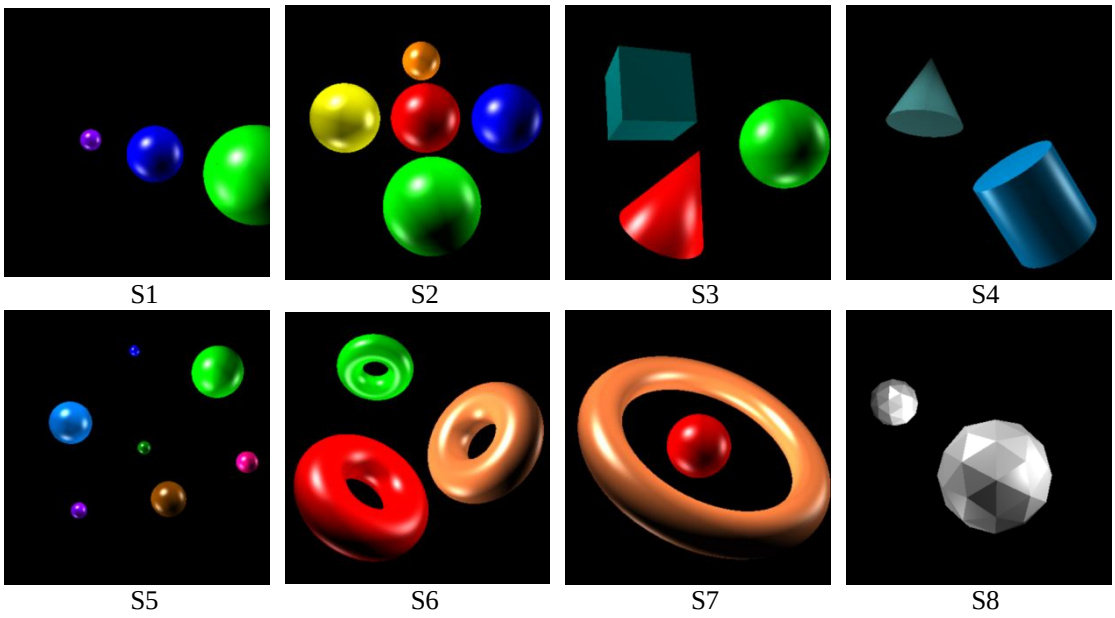


Figure A.2: Showing the 8 Synthetic images that used in the experiment; these images are created by graphics software.

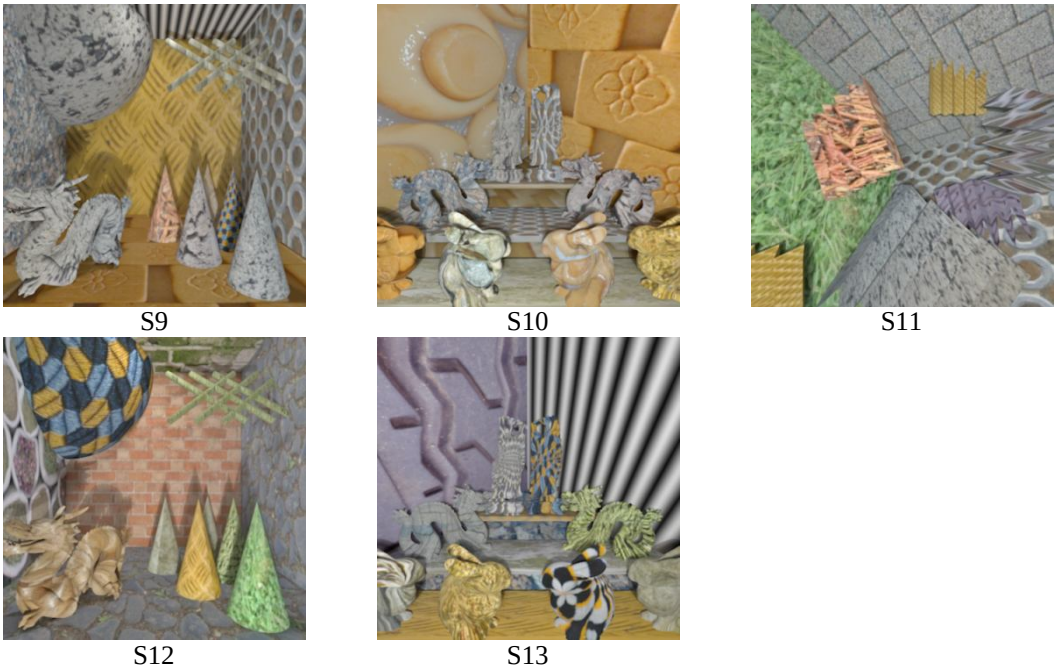


Figure A.3: Showing the 5 Synthetic images that used in the experiment; these images are downloaded from Alberta University.