DIGITAL WATERMARKING OF 3D DISPARITY MAP BASED IMAGES FOR TRANSACTION TRACKING

by

Ramzy Jaber Bachelor of Engineering, Ryerson, 2012

A thesis presented to Ryerson University

in partial fulfillment of the requirements for the degree of Master of Applied Science in the Program of Electrical and Computer Engineering

Toronto, Ontario, Canada, 2014 ©Ramzy Jaber, 2014

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including an required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Digital Watermarking Of 3D Images

Master of Applied Science 2014 Ramzy Jaber Electrical and Computer Engineering Ryerson University

In this thesis, the basics of disparity map and watermarking are reviewed extensively. In order to embed binary information into images, a 3D image watermarking system was proposed.. This embedded information was to survive the 3D Image rendering process of Disparity maps, to help identify malicious user who would distribute the watermarked image through an unauthorized system.

The proposed system adopted the concept of hidden pixel and introduced an algorithm that identifies all known hidden pixels within the image. This information is combined with the Disparity map to generate a hidden pixel disparity map (HPDM); using the information in the HPDM a decision matrix is generated. This decision matrix is used to guide the watermark embedding process to ensure that information embedded in the Left Image can survive the 3D rendering process.

Using the decision matrix, the watermark detector is capable of extracting the image from either the left or right image with no effect on the overall bit rate. This achievement is due to two original additions to the detection process: (1) Reverse rendering and (2) Cyclical Redundancy check. The proposed reverse rendering process expands the decision matrix into a reduced disparity map. This reduced disparity map is used to reverse the right image into a reduced left image. The identification of the image (left or right) is achieved through the use of a CRC check, which is also capable of detecting any errors in the extracted message, thus reducing the number of misidentification.

The proposed system was implemented and tested using MATLAB. The bit efficiency of the proposed system varied between 38% and 88%. This variance is caused by the complexity of the depth scene as well as the cost function used in the depth estimation process. The watermark embedding system proposed had a PSNR of 45 dB (when no mark was embedded); this value is primarily attributed to some of the quantization that occurs during the DCT transform. However, a PSNR of 33dB is attained when the watermark was added at full strength.

Acknowledgements

The completion of my thesis would not have been possible without the help of Dr. Kaamran Raahemifar. I would also like to thank Dr. Anastasios Venetsanopoulos, without his suggestions I would not have been able to explore such an increasing research topic.

I would also like to thank Dr. Mahmood Kassam and Dr. Gosha Zywno. With their trust, I was the lead teaching assistant for their courses. The responsibility I was entrusted with had many rewarding challenges that I will always be thankful for.

I also wish to thank Ghassem Tofighi, Ray Phan, Richard Rzeszutek, and Yasir Shoaib my labmates. For their companionship, advice, and support during the stressful periods of my degree. I would also like to thank Jason Naughton for lending me his insight in customizing my linux distribution at home.

Finally, I would like to thank my family for their continued support.

Dedication

I dedicate this to my friends and family.

Table of Contents

Chapter 1 Introduction	1
1.1 Motivation	1
1.2 3D Vision	
1.2.1 Human Binocular Vision	
1.2.2 Binocular Vision Image Capture	5
1.2.3 3D Display Technology	5
1.2.4 Image formats and Compression	9
1.3 Thesis Objectives	11
1.4 Thesis Contribution	
1.5 Outline	
Chapter 2 Stereo Correspondence	15
2.1 Disparity Maps vs. Depth	15
2.2 Image Rectification	17
2.2.1 Cost Function	19
2.3 Disparity Preprocessing	
2.4 Reconstruction	
Chapter 3 Background Theory	24
3.1 Watermarking models	
3.2 Watermarking Process	
3.2.1 Watermark Embedding	
3.2.2 Watermark Detection	
3.3 Watermark Pattern	
3.4 Image Processing Concepts	
3.4.1 DCT Transforms	
3.4.2 Quantization	
3.4.3 Zig-Zag Scan-lines	
Chapter 4 Literature Survey	
4.1 Stereoscopic Image Pair Watermarking	
4.2 Depth Map Image Watermarking	

4.3 Disparity Map Image Watermarking	40
4.4 Limitations of Existing Work	41
Chapter 5 Advanced Watermarking Concepts	
5.1 Watermark Attacks and Metrics	
5.2 Watermark Robustness	
5.2.1 Convolutional Code	
5.2.2 Scrambling	
5.3 Message Authentication	
Chapter 6 Proposed System	51
6.1 System Overview	51
6.2 Pre-processing	
6.2.1 Disparity Map	
6.2.2 Hidden Pixel Disparity map	
6.2.3 Message Processing	
6.3 Embedding Process	
6.3.1 Decision Matrix	
6.3.2 Embedding Process	
6.4 Detection Process	
6.4.1 Watermark Extraction Process	
6.4.2 3D Image Decoder	
6.5 Practical Realization of Rendering	60
6.6 Assumptions and Observations	
6.6.1 Assumptions	
6.6.2 Design Observations	64
6.7 Comparison with Other Work	
6.7.1 Watermark Characteristics	
6.7.2 Novelty of the Proposed System	
Chapter 7 Results and Evaluation	
7.1 Bit Capacity	
7.2 Fidelity	70
7.3 Robustness	71
7.3.1 Noise Attack	71

7.3.2 Filtering Attacks	73
7.3.3 Miscellaneous Attacks	74
Chapter 8 Conclusion and Future work	75
8.1 Summary	75
8.2 Future work	76
References	78

List of Figures

Figure 1.1: Transaction Tracking Client	2
Figure 1.2: Examples for (a) Diplopia Cause, and (b) Binocular Rivalry Cause	4
Figure 1.3: (a) Zero Parallax, (b) Negative Parallax, and (c) Positive Parallax	4
Figure 1.4: (a) Left Image (b) Buffer (L-R Transition) (c) Right Image (d) Buffer (R-L Tran	sition)6
Figure 1.5 Example of Passive 3D Display	7
Figure 1.6: Example of Autostereoscopic Display	8
Figure 1.7: (a) Stereo Pair, (b) Depth Based Image, (c) Disparity Map Image	9
Figure 2.1: Disparity-Depth Relation	16
Figure 2.2: (a) Camera view With Rotation (b) Equivalent Camera View After Rectification	Disparity
Map Objective Function	18
Figure 2.3: Disparity Optimization	19
Figure 3.1: Basic Watermarking Model	24
Figure 3.2: Standard Communication Watermark Model	25
Figure 3.3: Informed Communication Watermark Model	26
Figure 3.4: Vector Representation of Watermarking	
Figure 3.5: Vector Representation of Detection	
Figure 3.6: Zigzag Scan-line	34
Figure 5.1: (a) Block Diagram of Convolutional Code, (b) State Machine Representation of	
Convolutional Code	46
Figure 5.2: Viterbi Decoder Branching	47
Figure 5.3: CRC Code Generation	49
Figure 5.4: CRC Generation Flowchart	50
Figure 6.1: The proposed System Overview	51
Figure 6.2: Disparity Map Estimation Process	52
Figure 6.3: Hidden Pixel Example	53
Figure 6.4: Hidden Pixel Disparity Map Generation Flowchart	55
Figure 6.5: Embedded Message Format	56
Figure 6.6: (a) Hidden Pixel Disparity Map, (b) Decision Matrix	57
Figure 6.7: Embedding Process	58
Figure 6.8: Block Level Extraction Process	59

Figure 6.9: Realistic Rendering Flowchart	62
Figure 7.1: (a) Original Two Layer Image, (b) Rendered Two Layer Image	70

List of Tables

Table 1.1: Classification Based On Extraction	3
Table 6.1: Comparison of System Characteristics	67
Table 7.1: Bit Capacity of the Image	69
Table 7.2: PSNR Values	71
Table 7.3: Gaussian Attack Results	72
Table 7.4: Salt and Pepper Results Left Image	72
Table 7.5: Salt and Pepper Results Right Image	73
Table 7.6: Filter Attacks	74
Table 7.7: Miscellaneous Attacks	74

List of Algorithms

Algorithm 5.1: CRC Code Generation Algorithm	48
Algorithm 6.1: Hidden Pixel Disparity Map Generation Algorithm	54
Algorithm 6.2: Realistic Rendering	60

Glossary

Short Form	Full Form
3D	Three Dimensional
BBC	British Broadcasting Company
BER	Bit Error Rate
CC	Coefficient Correlation
CRC	Cyclical Redundancy Codes
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FrFT	Fractional Fourier Transform
GCD	Greatest common divisor
HPDM	Hidden Pixel Disparity Map
ID	Identification Number
LC	Linear Correlation
MSE	Mean Square Error
NC	Normal Correlation
PNG	Pseudo Noise Generator
PSNR	Peak Signal-To-Noise-Ratio
RGB	Red-Green-Blue
ROI	Region of Interest
SVD	Single Value Decomposition
UVW	Unseen Visible Watermark
YCbCr	Luma, Blue Difference, Red Difference.

Chapter 1 Introduction

1.1 Motivation

Three dimensional (3D) images and movies have been gaining popularity with mainstream crowds in recent years. This interest began in 2003, with the introduction of 3D movies. As the technology became more popular, 3D displays and multimedia systems were introduced into the marketplace. This medium continues to gain popularity as more movies are released in 3D. Recently, the British Broadcasting Company (BBC) released the 50th anniversary episode of its popular television series (Doctor Who) in 3D.

Today, the most active field in 3D display technology is the conversion of existing 2D media into the 3D format. However, 3D media has not yet penetrated the mass distribution networks, such as "Netflix" and "Hulu". It is likely that as its popularity increases over time and as 3D-content encoding technology improves, online movie networks will adopt 3D technology in the foreseeable future. When this occurs, to ensure the healthy growth of this technology the protection of copyright holders is imperative. While the protection of various types of copyright content is a popular research field, that of 3D media remains immature.

Several methods for copyright protection are available. Transaction tracking is the most used, because mass distribution of services occurs over Internet networks. "Safari Books Online" is a prime example. Whenever a user purchases an eBook, the transaction is embedded with hidden information that can identify the user. This encryption is done to determine whether or not a user is malicious. The user's information is redistributed on a peer-to-peer network. If the user is malicious, then he/she is identified and his/her access is terminated. As 3D media gain popularity and their distribution becomes more common, creating systems to protect their contents will become essential.

There are various applications of watermarking schemes [1], but copyright protection is of particular interest to this thesis. These schemes insert invisible watermarks (of a pseudo-random pattern) into digital media (e.g., Audio Files, Images, or Films) as binary or grey scale images, which are typically used to identify the copyright owner. This method is practical in applications where copyright ownership might be difficult to prove, such as a photograph of a popular location.

Transaction Tracking Systems are used to monitor and protect the copyright of material being transmitted from a database to a user, while embedding a unique user identifier (watermark). These systems are extremely useful when proof of ownership is not required, for example, a popular movie. Instead, these systems allow registered users to download copyrighted material. The acquired content usually contains identifiers that are unique to the users [2]. The advantage of transaction tracking is that service providers can easily distribute their content to end-users without worrying about malicious users. If users happen to be malicious and redistribute the content onto peer-to-peer networks, the copyright owner is able to identify them because of a unique identifier embedded into the media. Figure 1.1 illustrates the behavior of a transaction tracking system where a copyright owner provides contents to two distributors. The first distributor implements a transaction tracking system once a malicious user provides the content onto a peer-to-peer network. The distributor is capable of identifying the offending user.



Figure 1.1: Transaction Tracking Client

These schemes take one of the following forms: (1) Private Watermarking, (2) Semi-Private watermarking, and (3) Public watermarking (see [3]). The differences between the three approaches are defined in Table 1.1, where "Y" means "Yes, the data is required," "M" means "Maybe the data

will be required," and "N" means "No, the data is not required." Please note that the scheme proposed in this thesis is based on public watermarking.

Necessary Data	Private	Semi-Private	Public
Original Image	l Image Y N		Ν
Кеу	М	Y	N
Watermark Pattern (Pattern Key)	М	М	Y

Table 1.1:	Classification	Based On	Extraction
-------------------	----------------	-----------------	------------

1.2 3D Vision

1.2.1 Human Binocular Vision

While it is widely accepted that human eyes are capable of perceiving depth, the process by which the brain interprets the depth is still unknown. Due to the complexity of the human visual system, only the basics of this topic will be discussed here. Even though dual images aid in the process of depth estimation, they are not the only methods involved. Two examples are judgment from the size of known objects, such as cars, people, etc.and convergence of parallel lines, such as roads. While these approaches are easy to implement with human vision, they would be more difficult to perform computationally.

Hawkins [4] identifies some visual phenomena that are relevant to stereoscopic displays. The two images that are perceived by the left and right eyes are usually fused into a single image that contains non-quantitative depth information (stereopsis). If fusion of the two images fails, then one of these two phenomena is the cause: (1) Diplopia, in which two ghost images are created, and (2) Binocular rivalry, when an image from one eye is suppressed in favor of an image from the other eye, typically leading to oscillation between the two images. Diplopia is caused when eyes cannot converge onto an object, such as when an object is too close to the viewing center. Binocular rivalry occurs when each eye, receives a different image; where the two images cannot converge into a single view. Figure 1.2 (a) and (b) present an example of the occurrence of both events.



Figure 1.2: Examples for (a) Diplopia Cause, and (b) Binocular Rivalry Cause

The final component of interest for stereoscopic 3D displays is parallax. It is used to define the location of objects relative to the screen. Bruder et al. [5] explain how to create a 3D virtual environment where images of an object are created on the screen (zero parallax), out of the screen (negative parallax), and into the screen (positive parallax). These cases are illustrated in Figure 1.3 (a), (b) and (c), respectively. Note that the red dots indicate the left eye, and cyan dots represent the right eye.



Figure 1.3: (a) Zero Parallax, (b) Negative Parallax, and (c) Positive Parallax

1.2.2 Binocular Vision Image Capture

Humans perceive depth due to the disparate positions of an object's images created by the left and right eye. 3D Display technology uses these images to create an artificial depth on a single flat surface. This effect is achieved by providing the display with left and right images and projecting them to form a single image. Grinberg et al. [6] noted that two issues arise when dealing with simulated depth, which are as follows:

- 1. Image Acquisition: how an image is captured for each (left and right) eye,
- 2. Image Display Tool: how the captured images are isolated and displayed to the appropriate eye, which will be discussed in section 1.3.

Image Acquisition considers many aspects such as: (1) Correct distance from camera, (2) Angle between image capture planes, and (3) Lens offset. To choose the optimal parameters for each of the systems, the images should never be captured for a single viewing experience for they are usually displayed on various surfaces and viewed under various conditions. For that reason, a set of assumptions must be made to simplify the process and create a near-optimal viewing condition for the most common scenarios.

Note that the assumptions are meant for a single flat rectangular surface for display. The first assumption is that the display environment is the same as the recording environment. That is, the left and right images are simply a scaled version of the original recording. The second assumption is that both eyes are at the same distance from the display surface (i.e., the viewer is perpendicular to the screen). The last assumption is that the center of the viewer's eyes is at the center of the 3D display. Under these assumptions, the distance between the two cameras will be directly proportional to viewing conditions (distance between the user's eyes and the distance from the screen).

1.2.3 3D Display Technology

There are two categories for classifying 3D Display technology. The first category involves the method used to isolate the left and right images, while the second category involves the nature of the surface, which will not be included in this thesis. Flat rectangular surfaces are the most popular on the market, so the focus will be on this format. Interested readers can refer to [6] for further discussion of

the second category. The first display category is broken into the following subgroups: (1) Active, (2) Passive, (3) Autostereoscopic, and (4) Multiple-Screens.



Figure 1.4: (a) Left Image (b) Buffer (L-R Transition) (c) Right Image (d) Buffer (R-L Transition)

Active displays present the left and right images consecutively, thus ensuring that each eye observes the appropriate image and that the screen displays a black frame between the two images to isolate them [7]. During this period, the screen communicates with shutter glasses and informs it to close the shutter that is opposite to the eye it is about to display. An example of this cycle is presented in Figure 1.4, where (a) is the first cycle, (b) is the second, (c) is the third, and (d) is the fourth. After the fourth phase the monitor returns to the first. Unfortunately, the range of the viewing angle for these displays is limited; they are not meant for multiple users and are ideal for situations such as computer monitors, where the user is in a stationary position and only one user is operating the

system. One such example is the "Nvidia 3D Vision" system. The primary advantage of this system is the increased isolation between the left image and the right image, allowing for a stronger 3D effect.



Figure 1.5 Example of Passive 3D Display

Passive displays, unlike their active counterparts, display the left and right image simultaneously [7]. There are two primary versions of passive displays. The first is anaglyph 3D, where the left and right images are converted into a different color (left is converted to red, right is converted to cyan). Then, these images are combined and displayed as one. Through the use of colored glasses, the images are filtered and delivered to the appropriate eye. A more recent advancement of this technology depends on polarizing the light for the left eye and right eye. The drawback of this approach is that a complete isolation cannot be achieved. However, wider viewing angles are possible. This is the technology that is used primarily in movie theaters. An example of anaglyph stereo-displays is provided in Figure 1.5.



Figure 1.6: Example of Autostereoscopic Display

Autostereoscopic displays also present the left and right image simultaneously. However, unlike the previous two systems, glasses are unnecessary for this case. The most common versions of these displays present multiple views by interlacing the pixels of the two images. To ensure the delivery to the appropriate eye, a barrier is used. This barrier is placed between two interlaced pixels to ensure that the left pixels are blocked from the right eye, while right pixels are blocked from the left eye. This approach to 3D is illustrated in Figure 1.6. The primary disadvantage of this approach is the limited viewing distance and angle. This system is primarily used in environments where the user is close to the screen and the viewing angle is perpendicular to the display, such as "Nintendo 3DS." For a few more modern approaches to autostereoscopic displays, interested readers can refer to [8] and [9].

Multiple-Screen displays are not popular because of the limited products available. However, they offer the best isolation of images and are chiefly used in virtual reality applications. They rely on the user wearing head gear in which two individual displays are used to send an image to each eye. To ensure that the two screens are isolated, a physical barrier is used. These systems are very similar to Stereoscopes used by children; one such system is the "Oculus Rift."

Other forms of 3D displays have not been discussed in this thesis due to their lack of availability to users. The focus of this section is on 3D Display technology that is easily accessible

by users. Interested readers are advised to read [10], where an in-depth survey of the most modern 3D Display technology is provided.

1.2.4 Image formats and Compression

Any 3-D scene is captured using *N* number of cameras, where *N* is greater than one. In recent years, more advanced systems have been implemented where 8 or 16 (see [11]) cameras are used. Although research on advancing this technology and extending its applications is active, it is still a few years away from being introduced into the mass market. For that reason, this section is limited to Image formats where N = 2. Three main formats are used for presenting images: (1) Standard Image Pair, (2) Center Image and Depth map, and (3) Left/Right Image and disparity map, which are demonstrated in Figure 1.7.



Figure 1.7: (a) Stereo Pair, (b) Depth Based Image, (c) Disparity Map Image

Currently, there are three compression methods for 3D Image Pairs [4]. The first is JPS, where the left and right Images are merged side by side and then compressed using a standard JPEG compression. The main advantage of such an approach is that full resolution can be maintained at the cost of computational run-time and a larger file size. Next, is Interlaced GIF, where the images are vertically interlaced; maintaining the vertical and horizontal dimensions of the original image. The

main disadvantage of this approach is that half of the vertical pixel information is lost. The final compression method is Colored Anaglyph, where each image is converted to a color anaglyph and then combined. It is, then, compressed using standard formats. The main disadvantage of this last format is that color loss would yield poor results when compressed; however, it is ideal for anaglyph display technology.

To learn about a few more compression techniques for 3D Images, see [4] and [11]. While various methods exist, both sources agree that disparity maps and depth maps offer the best compression ratios. Disparity map rendering is a limited system and will require adjustments due to the variable nature of disparity vectors. The second disadvantage of disparity maps is that the user cannot adjust the 3D depth. However, the primary benefits of disparity maps are the larger compression ratio, when compared to depth maps. This is attributed to the fact that the depth values are captured for each image, with a range between 0 and 255.

On the other hand, a depth map image can modify the perceived depth on the user's side. Depth map were not chosen due to a few disadvantages. The first disadvantage is the larger data requirement. The second disadvantage is the increase of computational complexity [12]. In this thesis, disparity maps were used due to their extremely high compression ratios when compared to depth maps. This choice will be further discussed in chapter 1.5. Section 2.1 will explore the differences between the two image representations.

To compare the three formats when image compression is considered, assume that we have an RGB that is 360 by 280, or 100,800 pixels. For the first image format (Stereo Pair), the storage requirement is 604,800 bytes. For the second format (Disparity Maps) with a maximum value 15 (from the results presented in Chapter 6), a total of 302400 bytes are required for the image and 50400 bytes for the disparity map, for a total of 352800 bytes. Finally, the third format (Depth Map) requires 403200 bytes.

There are cases when the disparity values exceed the 255 limit, such as when the image has an extremely high resolution. An easy solution would be removing the DC biasing, and dividing the disparity values according to the GCD (greatest common divisor). The storage of the bias and GCD value would require only a few bytes. The added advantage of using disparity map-based rendering schemes is that the maximum disparity value cannot surpass 3% of the image size, if a comfortable viewing experience is desired [13].

1.3 Thesis Objectives

The 3D watermarking research considered various (N) image formats to protect copyright content. For N = 2 formats, the Stereoscopic image pair is the least popular, while depth estimation-based schemes are the most popular. The majority of the proposed schemes provide unique solutions to the problem; yet watermarking schemes are not extremely versatile. Thus, the objective of this thesis is to design a copyright protection scheme for 3D Image content (disparity Map-based representation). This system needs to be able to store user data into a 3D image to prevent malicious users from redistributing the material. The system must also meet the following objectives:

- Flexibility: While the work in 3D watermarking content is an emerging field, much of the work done offers unique solutions to the problem and is limited to the scheme that was chosen. A flexible framework would allow for the adoption of existing 2D watermarking scheme in order to benefit from the existing body of work.
- Robustness: Robustness is the system's ability to survive malicious attacks attempting to destroy the copyright content of digital material. Many watermark embedding schemes consider robustness in the embedding process. While the objective of this thesis is to create a framework that is independent of the embedding process, this should also be placed for robustness of the system.
- Quality: The image content that is to be protected must be visually appealing to the end user. For this reason, the intended system needs to minimize the number of visual artifacts to ensure a viewer-friendly experience. These visual artifacts can occur for various reasons. However, in a watermarking system the most common source of artifacts is the embedded watermark, which normally adds a sinusoidal noise to the image. Currently, 3D quality metrics for a watermarking system are limited. For this reason, the proposed system will use 2D quantitative metrics.
- Significant Bit Capacity: This system is meant to track images and identify the users; therefore, the system should be able to store a large data payload. This payload will vary based on the number of users held by the distribution network. The payload should be larger than 35 bits, or five times the world's population as a precaution. Ideally, each individual should be assigned

one identification number. However, to account for population increases and multiple accounts the number of identification numbers was chosen to be five times the world's current population.

1.4 Thesis Contribution

In this thesis, some of the existing works that have been performed on depth maps and disparity maps have been combined. In combining the various schemes, a more general framework was created that allows for many of the existing 2D watermarking schemes to be quickly and easily adapted to 3D images. To achieve this goal, the disparity map was combined with the hole-detection layer (layer of missing pixels); by means of a combined map, the system has the ability to evaluate the pixel survival and depth. Using this method, the ideal block locations for embedding are also identified and the 2D watermark scheme of choice can then be used. The drawback of the hole-detection layer is its inability to distinguish between occlusion, hidden pixels, and pixel mismatches. However, it does provide enough information for an informed decision. This map is called the Hidden Pixel Disparity Map, due to the data combining the disparity map while removing the Hidden Pixel disparity values.

The HPDM is then used to generate a decision matrix, which ensured that the selected block would survive the 3D rendering process. The decision matrix has two primary functions (1) Guide embedding process and (2) Reverse the rendering process. The first function, while not unique, is the first decision matrix in the area that limits the embedding process due to the rendering condition the block will undergo. However, the second function is a novel contribution. The reverse rendering process is used to convert the received right image into a left image. This is achieved through the expansion of the decision matrix into a reduced disparity map. To allow the system to identify whether or not the reverse rendering process is needed, a CRC integrity check is used. This CRC check determines whether the embedded message is valid. This helps the detector to determine if the left detection process or right detection process had the correct output.

The proposed framework should be capable of easily adopting existing 3D watermarking schemes. Due to the available depth information from the combined map, depth-based criteria can be added into the watermarking framework. This system can be expanded to prevent the encoder from embedding the information in the foreground. Such an expansion, of course, comes at the cost of bit storage ability, but it can be used to create a more user-friendly viewing experience. While, the

complexity of the various sub-systems varies, they can easily be replaced with newer and more complex models causing little or no effect on the rest of the system.

1.5 Outline

The remainder of the thesis is outlined below.

Chapter 2

This chapter will focus on the generation of the disparity map. The relationship between disparity maps and depth maps will be introduced. The thesis will continue through the discussion of the basic considerations necessary for effectively generating disparity maps, including the optimization process and the cost function process. It will also briefly discuss the reconstruction process for disparity maps.

Chapter 3

This chapter will focus on the basic concepts of watermarking and the image processing techniques that will be used. The focus will be on idealized systems to simplify the concepts introduced. Basic terminology will be introduced at the beginning, which will expand through the discussion of the embedding and detection process models, while the image processing concept introduced will be limited to the methods used by the proposed watermarking scheme discussed in Chapter 5.

Chapter 4

This chapter will present a brief overview of the 3D watermarking methods presented in the literature, while briefly discussing their limitations in the context of transaction tracking.

Chapter 5

This chapter will expand on the knowledge introduced in Chapter 3 and will discuss nonideal systems. The chapter will start by exploring design considerations that must be adopted when designing watermarking systems. It will continue with the description of various watermark attack techniques that will affect the detection rate.

Chapter 6

This chapter will focus on the introduction of the proposed watermarking scheme. It will merge the concepts introduced in Chapters 2-5 as well as discuss several constraints in place due to various design considerations.

Chapter 7

This chapter will focus on testing the proposed system. Many of the tests performed in this phase would have been introduced in Chapter 5. However, a few metrics will be added and described. These metrics will be unique to the system.

Chapter 8

This chapter will summarize the key aspects of this thesis, the scientific contribution, and the future research that can be conducted in this area.

Chapter 2 Stereo Correspondence

In Section 1.4 the various N = 2 stereo formats were discussed. While the depth map and disparity map are separate formats, they are closely related. This relationship is explored in Section 2.1. Since disparity maps are used in this thesis, the rest of this chapter is devoted to discussing them in some detail. In practice, it is extremely difficult to obtain images that are perfectly aligned on an axis; various factors, such as a camera misalignment on a 3D stand or when a camera takes the photos on a slight incline can lead to this issue. These factors make the disparity matching process extremely difficult, but they can be corrected through rectification (Section 2.2). Section 2.3 will focus on discussing the disparity search range and constraints that are placed on generating the disparity map. Since the output of the disparity map optimization process is not always smooth, Section 2.4 will introduce an approach to increase smoothness. Finally, Section 2.5 will briefly discuss the reconstruction process to convert the disparity map and reference frame into a standard stereoscopic pair, in which hole filling is normally used to create a better reconstruction. It is outside the scope of this thesis and plays a minor role in the watermarking process.

2.1 Disparity Maps vs. Depth

Before the relationship between the disparity map and depth can be discussed, we must first define the information that is stored in each image format. The disparity map stores the change of pixel coordinates between the reference frame (usually the left image) and the frame to be reconstructed (usually the right image). While the depth map contains the normalized depth information, where zero represents an unknown depths and the values between 1 and 255 represent depths to be scaled later on. These depth values are used to warp the center image (image between Left and Right Frame).



Figure 2.1: Disparity-Depth Relation

From the image formats represented in Figure 1.7 (b) and (c) it is difficult to identify the relationship between the two image formats. However, it can be observed that the depth map and disparity map are similar. The key difference between the two formats is that the provided images are from different locations. Figure 2.1 illustrates the similarities between the two images. It is important to note that the example in Figure 2.1 considers only a cross section on the x-axis. In that figure, *d* is the disparity value and z_p is defined as the depth value (this value would be later normalized to create a depth map). From the figure it is clear that the disparity value is defined as:

$$d = x_L - x_R \tag{2.1}$$

where, x_L is the coordinate of point $P(x_p, y_p, z_p)$ in the left image and x_R is the coordinate of point $P(x_p, y_p, z_p)$ in the right image. Using basic trigonometric relationships we obtain:

$$\frac{z_p + f}{z_p} = \frac{x_R - x_L + b}{b} \tag{2.2}$$

where, z_p is the depth of point $P(x_p, y_p, z_p)$, f is the focal length, and b is the baseline separation, which is defined as:

$$b = C_L - C_R \tag{2.3}$$

where, C_L is the center of the left image and C_R is the center of the right image. By substituting Equation 2.1 into Equation 2.2, *d* can be determined as:

$$d = \frac{b \cdot f}{z_P} \tag{2.4}$$

This representation is adopted from [14]. However, the rendering ability of the disparity maps is limited compared to the depth maps that are directly proportional to the value of z_p . This can be attributed to the lack of baseline and focal length information in the disparity value. For this reason disparity maps are capable of generating a single depth view, while depth maps are capable of generating multiple views.

2.2 Image Rectification

The rectification process is based on rotating the two received images. This action is performed to counteract the rotation of the camera pairs and to ensure that the axes of the two images are parallel (epipolar lines) while each point on the image stays on the same y-axis. This allows the search range of the disparity map estimation to be on the horizontal axis, significantly reducing the computational time of the disparity estimation process. The two views are rotated, resulting in a 2D coordinate change, which maintains the depth information [15] so that disparity values are preserved. It is important to note that the test sets used in this paper are already rectified. Interested readers are encouraged to read [16] and [15] for more details. It is important to note that the rectification process can be expanded to compensate for any rotations that the two cameras might experience. A 2D model of rectification is presented in Figure 2.2.



Figure 2.2: (a) Camera view With Rotation (b) Equivalent Camera View After Rectification Disparity Map Objective Function

Each pixel in the reference image is matched to a pixel in the disparate image. Due to the large number of pixels that exists within each image, this is a computationally intense process. The process of finding the disparity map usually requires a minimization process, such as the one defined as:

$$d(x, y) = \arg\min_{k \in S} C(x, y, k)$$
(2.5)

where d(x, y) is the disparity value at coordinate (x, y), k is the value to be optimized, S is the allowed search range of k, and C(x, y, k) is the cost function. Since this is a minimization process, the cost function is used to find the dissimilarity between the two images, where a higher value means a greater dissimilarity between the two pixels.

Let $I_L(u, v)$ be the pixel value of coordinate (u, v) of the left image, reference frame in this case, $I_R(u, v)$ be the pixel value of coordinate (u, v) of the right image, disparate frame. Let W(x, y, w) be the windowing function, and w be the number of pixels with the window. While the disparity value can take any value between 0 and the Image Size, it is more likely that the disparity value will be within 3 - 5 % of the image size [13]. To prevent any unnecessary computation, S is used as a way of reducing the search space. In this thesis the search range, S, is defined as:

$$S = \{s \in \mathbb{Z}^{\geq} | D_{min} < s < D_{max}\}$$

$$(2.6)$$

where D_{min} is the minimum disparity value, D_{max} is the maximum disparity value and *s* is the allowed disparity search range. An example of this process is presented in Figure 2.3.





2.2.1 Cost Function

It was previously mentioned that a cost function is used in the minimization process. Usually the cost functions in this research area operate as a measure of dissimilarity, where the closer the value is to zero the more the two windows are matched. Table 2.1 provides a list of some of the more common cost functions. Daniel Scharstein and Richard Szeliski provide a comprehensive analysis of the performance of various cost function (presented in [17]). The other problem faced with Equation 2.5 is that pixel uniqueness cannot be guaranteed. To combat this problem, a windowing function compares the neighboring pixels as well as the pixel of interest to reduce the rate of false matches. The windowing function is presented in Equation 2.7.

$$W(x, y, w) = \left\{ (u \cap v) \epsilon \mathbb{Z}^{\geq} \left| x - \frac{w}{2} \le u \le x + \frac{w}{2}; y - \frac{w}{2} \le v \le y + \frac{w}{2} \right\}$$
(2.7)

where (u, v) are the coordinates of the window surrounding the target pixel; w is the width or height of the square window; and (x, y) are the coordinates of the target pixel. In this thesis, five categories of cost functions were used: (1) Sum of Absolute Difference (SAD), (2) Sum of Gradient (SGRAD), (3) Sum of Squared Difference (SSD), (4) Normalized Cross Correlation (NCC), and (5) Mean Square Error (MSE). The SAD equation takes the absolute difference between pixels in the original block and the comparison block. These values are then summed together to form a value which represents the L^1 norm of the blocks. This function is defined below:

$$C(x, y, k) = \sum_{(u,v)\in(x, y, w)} |I_L(u, v) - I_R(u, v)|$$
(2.8)

The main disadvantage of this approach is the system's inability to account for lighting variance. This deficiency is improved by removing the mean of the blocks (Zero Mean SAD). The equation below defined the ZSAD metric.

$$C(x, y, k) = \sum_{(u,v)\in(x, y, w)} \left| I_L(u, v) - \overline{I_L(W)} - I_R(u - k, v) + \overline{I_R(W - (k, 0))} \right|$$
(2.9)

This problem is also addressed in the locally scaled SAD function, however this scaling is achieved through the normalization of the second block (right image block). Using the LS-SAD, the cosine of the angle between the two blocks in geometric space is used to evaluate the difference between them. This is describe in the equation below:

$$C(x, y, k) = \sum_{(u,v)\in(x,y,w)} \left| I_L(u,v) - \frac{\overline{I_L(W)}}{\overline{I_R(W - (k,0))}} I_R(u - k, v) \right|$$
(2.10)

The Sum of Squared Difference finds the sum of the squared value of this reflects the sum of the deviation of errors. However, this cost function emphasizes larger error allowing it to play a larger role in the equation; this characteristic allows the system to combat white Gaussian noise. This equation is defined below:

$$C(x, y, k) = \sum_{(u,v)\in(x,y,w)} |I_L(u,v) - I_R(u,v)|^2$$
(2.11)

Meanwhile, the Zero-Mean SSD finds the standard deviant of errors between the two blocks. This calculation is described in the equation below:

$$C(x, y, k) = \sum_{(u,v)\in(x, y, w)} \left| I_L(u, v) - \overline{I_L(W)} - I_R(u - k, v) + \overline{I_R(W - (k, 0))} \right|^2$$
(2.12)

The locally scaled SSD, however, measures the cosine of the angle between the squared difference in geometric space this is described in the following equation:

$$C(x, y, k) = \sum_{(u,v)\in(x,y,w)} \left| I_L(u,v) - \frac{\overline{I_L(W)}}{\overline{I_R(W - (k,0))}} I_R(u - k, v) \right|^2$$
(2.13)

The MSE, on the other hand, is similar to the ZSSD proposed in equation 2.12; however, the sum is scaled relative to the size of the window, as presented in the equation below:

$$C(x, y, k) = \sum_{(u,v)\in(x,y,w)} \frac{\left|I_L(u,v) - \overline{I_L(W)} - I_R(u-k,v) + \overline{I_R(W-(k,0))}\right|^2}{w^2}$$
(2.14)

In contrast, the Sum of Absolute Gradient Difference uses the gradient of the image to determine the similarity between the blocks. By using the gradient, the equation emphasizes the comparison of the edges of the objects (structure of the image) rather than the textures. This equation is presented below:

$$C(x, y, k) = \sum_{(u,v)\in(x,y,w)} |\nabla_x I_L(u,v) - \nabla_x I_R(u-k,v)| + \sum_{(u,v)\in(x,y,w)} |\nabla_y I_L(u,v) - \nabla_y I_R(u-k,v)|$$
(2.15)

The final cost function class is the NCC. The NCC is a metric to find the similarity between two signals. Unlike the SAD and the SSD, the value of the NCC is already normalized. The equation is defined below:

$$C(x, y, k) = \frac{\sum_{(u,v)\in(x,y,w)} I_L(u,v) \cdot I_R(u-k,v)}{\sqrt{(\sum_{(u,v)\in(x,y,w)} I_L(u,v)^2) \cdot (\sum_{(u,v)\in(x,y,w)} I_R(u-k,v)^2)}}$$
(2.16)

To aid in the system's robustness against variations in the luminance level, the zero mean NNC is used. This equation is defined below:

$$C(x, y, k)$$

$$= \frac{\sum_{(u,v)\in(x,y,w)} \left(I_L(u,v) - \overline{I_L(W)} \right) \cdot \left(I_R(u-k,v) - \overline{I_R(W-(k,0))} \right)}{\sqrt{\left(\sum_{(u,v)\in(x,y,w)} \left(I_L(u,v) - \overline{I_L(W)} \right)^2 \right) \cdot \left(\sum_{(u,v)\in(x,y,w)} \left(I_R(u-k,v) - \overline{I_R(W-(k,0))} \right)^2 \right)}}$$

$$(2.17)$$

2.3 Disparity Preprocessing

One of the side effects of the optimization process presented in Section 2.2 is that the generated disparity maps are not smooth. To counter this problem, the following operations have to be performed: (1) Cropping, (2) Disparity Regularization and (3) Median Filtering. While the cropping process is not always discussed in the literature, it will be discussed here. Due to the windowing function and the disparity shift, the disparity can be estimated for only a specific region in the image because the two values create boundaries. This boundary is defined as:

$$B = \begin{cases} \left\{ x \in \mathbb{Z}^{\geq} \middle| D_{min} + \frac{w}{2} < x < x_{max} - \frac{w}{2} \right\}, & \text{if } x \\ \left\{ y \in \mathbb{Z}^{\geq} \middle| \frac{w}{2} < y < y_{max} - \frac{w}{2} \right\}, & \text{if } y \end{cases}$$
(2.19)

where x_{max} is the maximum value for the *x* coordinate in the image and y_{max} is the maximum value for the *y* coordinate in the image. Using the method proposed in [18] and [19], the regularization of the disparity map is performed; to counter the effects of the boundaries, the image is cropped. This process can be defined in Equation 2.20:

$$D_{reg}(x,y)$$

$$= \begin{cases} \frac{1}{w} \sum_{(u,v) \in W(x,y,w)} D(x,y), & \text{if } \left| D(x,y) - \frac{1}{w} \sum_{(u,v) \in W(x,y,w)} D(x,y) \right| < T \\ D(x,y), & \text{otherwise} \end{cases}$$

$$(2.20)$$

where D(x, y) is the original disparity map, $D_{reg}(x, y)$ is the regularized disparity map and T is a threshold value. The advantage of such an equation is that false disparity values can be removed. This
neighborhood averaging-based scheme helps prevent misapplication of disparity values, while the threshold should provide better edge preservation. Finally, the disparity map is passed through a median filter to ensure that the system remains intact.

2.4 Reconstruction

$$P_{R} = \begin{bmatrix} x_{R} \\ y_{R} \end{bmatrix} = \begin{bmatrix} x_{L} - D(x_{L}, y_{L}) \\ y_{L} \end{bmatrix} = \begin{bmatrix} x_{L} \\ y_{L} \end{bmatrix} - \begin{bmatrix} D(x_{L}, y_{L}) \\ 0 \end{bmatrix} = P_{L} - \begin{bmatrix} D(x_{L}, y_{L}) \\ 0 \end{bmatrix}$$
(2.21)

The reconstruction of the right image from the left image and disparity map is a rather simple process. This process is shown in Equation 2.21, where P_R is the pixel coordinate in the right image and P_L is the coordinate in the right image. However, the biggest limitation of the equation is that it does not compensate for overlapping pixel locations. To combat the problem, Equation 2.21 will be subject to the condition presented in Equation 2.22. If the condition is not met, the system moves on to the next coordinate and ignores the current pixel.

$$\{[x_L - D(x_L, y_L)] \notin S_X | S_X \text{ is the set of all previous } x_R \text{ values}\}$$
(2.22)

Chapter 3 Background Theory

In this section, the basics of watermarking systems and image processing are discussed in four subsections. Section 3.1 provides an overview of the two models used by these systems. Embedding and detection processes are discussed in Section 3.2. Section 3.3 briefly discusses the generation of watermark patterns. Lastly, Section 3.4 briefly describes the image processing methods used in the literature, but the focuses of the section will be image-processing conversions.

First, there are few strict definitions that need to be stated to help the reader understand the contents of this section:

- (1) Coverwork defines the form of media that is going to undergo the watermarking process,
- (2) Watermarked Coverwork refers to the work that has already undergone the watermarking process,
- (3) Added Pattern is a pattern added to the work,
- (4) Reference Pattern is a pattern used for comparison during watermark detection, where (under ideal conditions) the reference pattern and the added pattern should be identical, and
- (5) The Message is encoded into the watermarked coverwork by the added pattern. A typical watermarking system is depicted in Figure 3.1.



Figure 3.1: Basic Watermarking Model

3.1 Watermarking models

The most basic model used in any watermarking system is presented in Figure 3.1. The figure is effective in illustrating how a watermarking scheme operates: a message is added to a work and then transmitted. Once a message arrives, it is decoded. While the model is basic and not used in practice, it is helpful in demonstrating some of the earlier definitions. This section describes two useful models for watermarking found in [20]. The first is a communication model. It is commonly used to describe the behavior of the system. An illustration of the model is provided in Figure 3.2. In this figure the dashed lines are components of the model that may or may not exist. These models correspond to watermarking schemes defined in Table 1.1.



Figure 3.2: Standard Communication Watermark Model

The model presented in the above figure assumes that the message is first sent to the encoder and then is mapped onto a watermark pattern layer. This layer is usually of the same type and size as the intended region of the coverwork. Afterwards, a watermark pattern layer is added directly to the coverwork and is then transmitted. Under ideal situations, the transmitted coverwork arrives intact and free of alterations.

However, during the transmission phase, the coverwork could be altered because of the channel characteristics or an attack by a malicious user. Various types of attacks are described in Section 5. For simplicity, the developed model assumes that an attack has two elements, namely, an

attack block (circular element with A) and an attack signal. For example, in the case of compression, the attack block would be the compression algorithm, while the attack signal would be the compression parameters. Afterward, the signal is sent into the detector at the receiver's end. The decoder generates a best guess for the embedded message, and outputs the message. While this communication model fits many watermarking schemes, it does not account for an informed embedder. An informed embedder accepts the coverwork as an input in the encoding stage and uses some of the information from the coverwork to determine the embedding process. Figure 3.3 illustrates such a system.



Figure 3.3: Informed Communication Watermark Model

The models provided above are useful in testing watermarking schemes under different attacks. However, they are poor when it comes to describing the embedding and detection processes. For these two processes, a geometric model is used to quantify the behavior. Each vector in this domain corresponds to a coverwork (an image). This is defined as the media space. What this suggests is that every image is an N-Dimensional Vector where each pixel is a dimension. The media space is a discrete rectilinear lattice, where the quantization of images is small enough that the space is treated as a continuous rectilinear lattice.

The media space model is used when considering embedding and detection. This model is used in Sections 3.2 and 3.3, but three necessary terms are defined here. First, Region of Acceptable fidelity is an N-Dimensional hypersphere centered on the coverwork vector where the image distortions are within acceptable quality limits. Second, Embedding Disruption is the total distortion that the system undergoes due to the embedding process. Finally, Detection Regions occur within the media space where the detector is able to identify a watermark within the message.

3.2 Watermarking Process

While some watermarking schemes add the watermark directly into the provided image, some transform the image into a different domain. The domain transform is usually referred to as the marking space. The marking space is typically used to simplify the watermarking process or to increase the system's ability to detect the watermark. The two popular marking spaces are spatial and transform domains. Spatial domain watermarking scheme relies on altering the provided pixel information to store the information in such a way that the overall effect on the system is undetected. An example of this is using the Least Significant Bits to store information. However, transform domain methods transform the pixels into a different representation, whereby the watermark is added into that domain and the process in inverted, for example, a DCT transform.

In this thesis, transform domain embedding is used despite the merits of both systems. While spatial methods provide a higher storage capacity and embed the information at a lower computational cost, they are also extremely fragile and cannot survive attacks, to meet the robustness objective transform domain watermarking was selected.

To simplify the watermarking analysis in sections 3.2.1 and 3.2.2, it will be assumed that the provided coverwork is already in its marking space (under went a domain transformation). This thesis also assumes that the transform process follows the Discrete Cosine Transform (DCT). DCT is the transform of choice for many reasons. It provides DC, low frequency, mid frequency and high frequency elements. This allows the ability to balance the fidelity and robustness constraints, by selecting the embedding region. When embedding within DCT marking space, low frequency watermark insertion allows for higher resistance to attacks, but it suffers from a lower image quality. In contrast, embedding in the high frequency regions provides a lower resistance to attacks but yields a higher image quality. To achieve a balance between the two, in this thesis, the watermark is added to the middle frequency bands as in [21]. Subsection 3.2.1 introduces the watermarking addition process and assumes that the provided information is already converted to its marking space (DCT). In Sub-section 3.2.2, the detection process is discussed.

3.2.1 Watermark Embedding



Figure 3.4: Vector Representation of Watermarking

Now, assume a two-value coverwork is to be watermarked, represented by the pink and blue vectors in Figure 3.4. This representation treats marking space of p values as a p-dimensional vector. Afterwards, a watermark is added to the result (added pattern) of the two vectors, the watermark (red) vector. The addition of the two vectors is given as:

$$I_m = I_0 + \alpha w_{pf} \tag{3.1}$$

where I_m is the watermarked coverwork; I_0 is the coverwork and α is a scaling factor to increase the watermark pattern strength. The watermark addition introduced above is the most basic form of watermarking. The disadvantage of Equation 3.1 is that it exists in two states: (1) watermarked, and (2) non-watermarked. This approach makes it difficult to store significant information. It was mentioned in Section 3.1 that the pattern is regularly used to encode a message. This encryption is achieved through the use of the following equation:

$$I_m = I_0 + \alpha b w_{pf} \tag{3.2}$$

where *b* is the bit that is being embedded (-1 for a value of 0 and 1 for a value of 1). Equation 3.2 allows for three states: (1) Negative Watermark, (2) Positive Watermark, and (3) No Watermark. This thesis makes use of the addition described in Equation 3.2.

3.2.2 Watermark Detection

While many different approaches to watermark detection exist, this thesis opted for correlation-based detection because it minimizes the need for input files to be present. Normally, after a correlation is performed on the marked coverwork, the value of the correlation is compared to a predetermined threshold value (T), to determine the bit that is stored. The conditions are presented as:

$$f_{d}(I_{m}, w_{pf}) = \begin{cases} No \ Decision & \text{if } |z(I_{m}, w_{pf})| < T \\ 1 & \text{if } z(I_{m}, w_{pf}) > T \\ 0 & \text{if } z(I_{m}, w_{pf}) < -T \end{cases}$$
(3.3)

where $z(I_m, w_{pf})$ is the correlation function. There are three distinct correlation functions that are effective in this scenario. The first two are linear correlation and normalized correlation, respectively. These functions will be used to describe the detection process using the geometric model. The third function is the coefficient correlation, which is the correlation used in this thesis. Linear correlation (LC) is defined as:

$$Z_{LC}(I_m, w_{pf}) = \frac{1}{p} \sum_{\forall i} I_m[i] w_{pf}[i]$$
(3.4)

where p is the size of the provided vector. It is clear from Equation 3.4 that the value of the correlation is the dot product of the two vectors. If w_{pf} is constant, the constant value can be interpreted as an orthogonal projection of the coverwork vector onto the reference pattern vector. When compared to a threshold value, the set of vectors that exceed the threshold value will be to one side of a plane that is perpendicular to w_{pf} . This boundary is a (p - 1) hyperplane, as illustrated in Figure 3.5. For simplicity, the detection process in the presented figure assumes that a two-state embedding process is used (Equation 3.1). Unfortunately, this detection mechanism has a large detection region that increases the rate of false detections.



Figure 3.5: Vector Representation of Detection

The second correlation approach used is the normalized correlation, which is defined in Equation 3.5. The primary benefit of this approach is that the system removes the correlation's dependence on magnitude. In other words, changes in brightness or contrast have a reduced effect on the detection process.

$$Z_{NC}(I_m, w_{pf}) = \sum_{i \in x} \frac{I_m[i] \cdot w_{pf}[i]}{|I_m| \cdot |w_{pf}|}$$
(3.5)

Equation 3.5 can be interpreted as a dot product operation. This equivalence is presented in the equation below:

$$I_m \cdot w_{pf} = |I_m| \cdot |w_{pf}| \cdot \cos\theta \tag{3.6}$$

Equation 3.6 suggests that the decision boundary for the NC correlation is a (p - 1) dimensional hyper-cone when the reference pattern is orthonormal to the coverwork. The image itself will play no role in the detection process. The equation below (3.7) illustrates the results of the correlation if the reference pattern is orthonormal to the coverwork.

$$I_m \cdot w_{pf} = (I_0 + \alpha w_{pf}) \cdot w_{pf} = \alpha w_{pf} \cdot w_{pf}$$
(3.7)

The method presented in Equation 3.8 is effective in addressing mean shifting and contrast enhancement that LC correlation struggles with. To fully remove the sensitivity to mean shifting, coefficient correlation (CC) is used in this thesis, presented as:

$$Z_{CC}(I_m, w_{pf}) = \sum_{i \in x} \frac{(I_m[i] - \overline{I_m}) \cdot (w_{pf}[i] - \overline{w_{pf}})}{|I_m - \overline{I_m}| \cdot |w_{pf}[i] - \overline{w_{pf}}|}$$
(3.8)

where, $\overline{I_m}$ is the mean of the image block and $\overline{w_{pf}}$ is the mean of the pattern. This scheme addresses the weakness of the previously mentioned schemes. Therefore, it is the CC correlation used in this thesis.

3.3 Watermark Pattern

The majority of watermark pattern generation schemes rely on two components: (1) Key and (2) Pseudo Noise Generator (PNG). Normally, the key value is send to PNG as an initial seed; hence, the output of the PNG is semi-predictable. With this added key, the receiver can recreate the reference pattern with a single value. This approach is heavily influenced by the Spread Spectrum Communication. In SSC, a narrow baseband signal is spread across multiple frequencies such that an attack is more difficult to detect. This spreading is usually performed with pseudo-noise sequences that are generated from PNG [22]. Ideally, the output of the PN sequence should have a zero mean and a unit standard deviation.

The spreading sequence mentioned above suffers from low energy levels. As the required sequence increases in size, the energy spreads across the p values. To combat this issue, this thesis uses the pseudo-random binary sequences. Various binary sequences exist within SSC. In these sequences the key is used to determine the initial state of the linear feedback shift registers. The primary properties of PN binary sequences are:

- Balance: A sequence of $(2^n 1)$ bits, should have (2^{n-1}) ones and $(2^{n-1} 1)$ zeros, and
- Autocorrelation: Autocorrelation of sequence should be 1, whenever i = 0, and $\frac{-1}{N}$ otherwise.

These properties make PN binary sequences very useful when using SSC. While there are various PN binary sequences, this thesis uses m-sequences. The rationale is because the watermark

embedding process used in this thesis requires one watermark pattern. The simplest method of PN generation that meets the needs of this thesis is the m-sequence generator. As for the LFSR polynomials, a polynomial listed in [23] was used.

3.4 Image Processing Concepts

In this thesis, some image processing methods are used to aid in the watermarking process. The discussion in this section focuses on format conversion. (Some image filtering techniques also used in this thesis are discussed in Section 5.) Prior to converting an image into its media space, the color format is changed from the RGB spectrum to YCbCr. This change is performed because the Transform Domain function (DCT), discussed in Subsection 3.4.1, converts one layer. The Y channel contains the luma information, and creates a lower level of distortion when the watermark was embedded in this domain. The Y-channel is treated as the coverwork in this paper. Section 3.4.2 discusses the quantization method that is used to increase the watermark's ability to survive attacks. Finally, Subsection 3.4.2 will discuss the Zig-Zag scanline, which was used to order the output of the DCT from low to high frequency components.

3.4.1 DCT Transforms

The DCT is a popular frequency transform that is commonly used in image compression. The most popular compression method that uses this transform is JPEG compression. The transform is a special subset of the DFT, where the phase information is discarded in favor of the amplitude information. The reason this transform was chosen in this system is due to its use with compression algorithms. Since compression is one of the most common watermark attacks, it allows us to counteract some of the effects. The following equation defines the DCT transform:

$$F(u,v) = \tag{3.9}$$

$$\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \alpha(u) \cdot \alpha(v) \cdot \cos\left[\frac{\pi(2 \cdot x + 1)u}{2M}\right] \cdot \cos\left[\frac{\pi(2 \cdot y + 1)v}{2M}\right] \cdot f(x, y)$$

where (u, v) are the frequency coordinates of the DCT transform, (x, y) are the pixel coordinates of the provided image, f(x, y) is the image, $\alpha(u/v)$ are scalar units define in Equation 3.10, and *M* is the block size.

$$\alpha(u/v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u/v = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases}$$
(3.10)

The inverse DCT function is defined below:

$$f(x,y) = \tag{3.11}$$

$$\sum_{u=0}^{M-1} \sum_{v=0}^{M-1} \alpha(u) \cdot \alpha(v) \cdot \cos\left[\frac{\pi(2 \cdot x+1)u}{2M}\right] \cdot \cos\left[\frac{\pi(2 \cdot y+1)v}{2M}\right] \cdot F(u,v)$$

The provided DCT and inverse DCT equations are a special case of the standard DCT. Normally, the DCT can accept asymmetrical block sizes where the number of elements in the x and the number of elements in the y direction are not equal. However, in this thesis only symmetrical sized blocks are used. For that reason the DCT equation reflects the intended usage.

3.4.2 Quantization

Quantization is a necessary step for the watermarking scheme in this thesis. It is performed to increase robustness to JPEG compression attacks. After DCT transform is performed, the DCT steps are quantized as defined as:

$$I_{bq} = q \cdot \left[\frac{I_b}{q}\right] \tag{3.11}$$

where I_{bq} is the quantized DCT block and q is the quantization matrix. However, it is important to note that the division and the multiplication in this equation are point by point. Thus, the quantization operation occurs according to elements and not to matrix. The quantization matrix that was used in

this algorithm is the same one used for JPEG compression and is provided below. In this thesis the intended block sizes are 8 by 8.

г16	11	10	16	24	40	51	61 -
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
L72	92	95	98	112	100	103	99 -

3.4.3 Zig-Zag Scan-lines

Zig-Zag Distortion is a form of rearranging the two dimensional DCT transform from the lowest frequency band to the highest frequency band. The purpose of this transform is the reduction of space necessary for JPEG compression. However, by converting the DCT transform of the image into a one dimensional vector, it becomes easier to embed the watermark into the middle frequency bands of the image. Thus, it can aid in finding the perfect balance between survivability and distortion (mentioned in Section 3.2). An example of Zig-Zag scan-lines is illustrated in Figure 3.6,



Figure 3.6: Zigzag Scan-line

Chapter 4 Literature Survey

In this chapter, an overview of 3D watermarking will be presented. The watermarking systems for the three image formats presented in Subsection 1.2.4 will be explored further. Section 4.1 will present some of the standard stereoscopic image pair watermarking schemes from the literature. Unfortunately, this is the least popular of the three formats, so the information presented will be limited. Section 4.2 will present the various watermarking schemes for depth map-based images. Finally, Section 4.3 will focus on watermarking schemes for disparity-based images, the format used in this thesis.

4.1 Stereoscopic Image Pair Watermarking

The work in [24] proposes that embedding decisions should be based on the smoothness of the image. To embed the information (binary message), both images are divided into 8 by 8 non-overlapping blocks; these blocks are then converted to the DCT marking space. Let $B_L(n)$ and $B_R(n)$ represent the n^{th} block for the left and right image, respectively. Once the blocks are in their marking space (frequency domain), the total energy of each block is calculated (for both sets). Using the calculated energy value, a decision is made regarding the smoothness of each block by comparing the energy level of the block to the average energy level of all blocks (both images). If the value is higher, the value is considered non-smooth, but otherwise it is considered smooth. Using the smoothness of the blocks, vector R is constructed. By comparing the smoothness decision of $B_L(n)$ and $B_R(n)$ a value of one or zero is stored in R(n), where a value of one is stored if the two blocks share the same smoothness decision, and zero if they do not. Upon the completion of the R(n) matrix the watermarking process begins. If the value in the R matrix matches the binary value to be stored, then the blocks are not modified and a value of zero is stored in M(n), a matrix responsible for tracking modifications to the image. If the value in R(n) is zero, the DC coefficient is quantized using the following formula: $Q = \left| \frac{DC}{S} \right|$. If the value of Q is odd, the following formula is used to find the new DC coefficient value: = $Q \cdot S + (1 - b) \cdot Q$, where b is the value to be stored. Otherwise, the following formula is used: $DC = Q \cdot S + b \cdot Q$. If the first block was quantized, a value of one is stored in M(n); otherwise, a value of zero is stored. Once all the blocks have been processed, an

inverse DCT is performed so that the blocks are recombined into full images. The modification matrix, M(n), and the parameter S are stored and would be used in the future to reverse the watermark and prove ownership of the image.

The work presented in [24] relied on the relationship between the two images to embed its information. This approach is similar to the work presented in [25], where the relationship between the two frames is used for embedding as well as the depth information between the images. The watermarking system that is proposed begins by finding the global disparity. The global disparity is similar to the local disparity, introduced in Section 2. However, we are primarily interested in the mode of the disparity map. Figure 2.2 (b) illustrates the typical point of view of two cameras. The field of vision of each camera is represented by dotted lines. From Figure 2.2 (b) it is clear that the left pixels of the left view and the right pixels of the right view are unique to each view and are not shared by the cameras. The global disparity is used to identify these regions. This value is stored as a key (k_1); afterwards the luminance layer of the shared region is divided into 8 by 8 blocks to once again undergo a marking space transform into the DCT domain. The Watson vision model is used to determine the strength of quantization error for the left and right images w_L and w_R . For added security, the message to be embedded is scrambled and stored into a single DCT coefficient (2^{nd} row, 3^{rd} column). The bits are embedded into the images, using the following equations if $|L - R| \le 0$:

(1) $R' = R + a_1 \cdot |L - R| - c_1 \cdot w_R \cdot (2b - 1)$

(2)
$$L' = L + a_2 \cdot |L - R| - c_2 \cdot w_R \cdot (2b - 1)$$

Otherwise, the following equations are used:

(1)
$$R' = R + a_3 \cdot |L - R| - c_3 \cdot w_R \cdot (2b - 1)$$

(2)
$$L' = L + a_4 \cdot |L - R| - c_4 \cdot w_R \cdot (2b - 1)$$

where *a* and *c* are weighted coefficients selected to ensure that the value of L' > R' if the bit to be stored (*b*) is 1. The work [26] uses a quantized disparity map to strip the image into three separate layers (background, intermediate and foreground). This layering process is performed by quantifying the extracted disparity map into three levels. The disparity map is extracted using the LL (corresponding to the image that undergoes Low-Pass, Decimation, and then Low-Pass) sub-bands of the left and right image. These sub-bands are obtained when the images are transformed using the discrete wavelet transform (DWT). Finally, the watermark is embedded into the HL (corresponding to the image that undergoes High-Pass, Decimation, and then Low-Pass), LH (correspond to image that undergoes Low-Pass, Decimation, and then High-Pass) and HH (correspond to image that undergoes High-Pass, Decimation, and then High-Pass) sub-bands of the right image. The embedding processes consist of using the quantized disparity map to create three separate masks, where each mask corresponds to a different sub-band of the DWT. Using the masks, certain regions of the sub-bands are omitted. The remaining regions of the sub-bands are then quantized using one of the following uniform quantizers:

(1)
$$Q_0 = \left|\frac{s_i}{\Delta}\right| \cdot \Delta + d$$

(2) $Q_1 = \left|\frac{s_i \cdot m_i}{\Delta}\right| \cdot \Delta + \frac{\Delta}{2} + d$

where Q_{\square} is the quantizer for bit 0 or bit 1, s_i is an unmasked coefficient from the DWT sub-band (LH, HL, and HH), Δ is the quantization step, and d is a key for added security. After the embedding process the sub-band masking is reversed, followed by the reversal of the DWT process. This system embeds the information solely in the right image. The extraction process of this scheme is similar to the embedding process; however, after the sub-band masking, the system attempts to identify which quantization process was used and makes a bit decision. This analysis is performed by passing the extracted coefficient through Q_0 ; the output of the quantization is subtracted from the extracted vector. If the absolute value is below $\frac{\Delta}{4}$ then the Q_0 quantizer was used. Otherwise, the Q_1 quantizer was used.

4.2 Depth Map Image Watermarking

The main challenge in developing watermarks for depth-based image formats is the need for the watermark to survive 3D warping. The 3D warping process shifts pixels along the horizontal axis to synthesize two images such that a left and right image are created with a sense of depth. Due to these horizontal shifts, there is no guarantee that the watermark will survive in the rendered images. To combat this problem, [27] proposes a new methodology that embeds the watermark in the noise of the image. This watermarking scheme exploits two principles: (1) Histogram of image noise has a zero mean Laplacian distribution, and (2) 3D Warping of Center image shifts information only

horizontally. The embedding process begins through the extraction of the noise image from the center view. Afterward, the extracted noise layer is split horizontally into k noise segments. The generated watermark pattern is a Gaussian zero mean distribution. Next, the mean of the Gaussian is shifted based on the element of the watermark that needs to be embedded, positive shift for a value of 1, and a negative shift otherwise. The embedding process can be defined by the following equation: $SW_i(x, y) = (N^{(0,c)}(x, y) + w_r(i)) \cdot \psi(x, y)$, where ψ is the scaling factor, which is dependent on the noise visibility and noticeable depth difference. The shifted noise vector will be added to the extracted noise, and the resultant noise vector is reintroduced into the de-noised image.

In [28] a different 3D video watermarking scheme is presented, which exploits some of the depth map characteristics. The first principle that it makes use of is the hidden pixel. When the left and right view images are rendered, there is a possibility that pixels will occupy the same location. This watermarking scheme identifies these locations and treats them as highly weighted embedding positions. It also considers some unique properties unique to 3D video content (z-motion), which will be omitted in the current discussion. The embedding process in this thesis uses spatial embedding (briefly discussed in Subsection 3.2.1). This relationship can be defined through the following equation: $F_t^w(x, y) = F_t(x, y) + \alpha_t(x, y) \cdot w_t(x, y)$. The weight function in this system is not constant, for its value varies based on a Human Visual Model, Hidden Pixels and the z-motion.

On the other hand, [29] solved the watermarking problem by embedding the watermark in the Left Image, Center Image and Right Image. The system proposed in this [29] creates three orthogonal watermark patterns, which will each be embedded in a single view (Left, Center, or Right). The system begins by splitting the image into *N* by *N* blocks. Each block is then selected, and two views are rendered (Left and Right). If the number of holes in the image are below a predetermined threshold (holes are a side effect of hidden pixels), then the block is embedded with information. The embedding process begins by embedding the center block with the center reference pattern based on the following equation: $= s + (\alpha \cdot b - \lambda \cdot s') \cdot w_{patternView}$, where s' is a scaling factor.

After the center block is embedded, the left view is re-synthesized from the watermarked center block and embedded using the same formula but using the left watermark pattern. The rendering process is reversed so that a double watermarked image center now exists. Finally, the right view is synthesized and once again watermarked with the right pattern; the rendering process is then reversed. The result is a system with three watermarks that will survive the rendering process.

Reference [30] presents a truly unique watermarking scheme based on the principle of unseen visible watermarking (UVW). The principle behind UVW is similar to watermarking in the real world. Under normal viewing conditions the embedded watermark is invisible. However whenever the viewing condition is changed the watermark becomes visible. A simple example of this is invisible ink, where under normal viewing conditions the ink cannot be seen. However, when it is exposed to UV light, the watermark becomes visible. The watermark proposed in [30] embeds an image invisible during standard rendering conditions. However, when one of the rendering parameters is changed, the watermark becomes visible. The equation used for rendering is defined by $= f \cdot t_x \cdot \frac{l_z}{255} \left(\frac{1}{z_{near}} - \frac{1}{255}\right)$

 $\frac{1}{z_{far}} + \frac{1}{z_{far}}$, the standard equation for all depth-based rendering. Through the addition of a small modification to I_z , which can be defined as ΔI , the equation becomes $d + \Delta d = f \cdot t_x \cdot \frac{I_z + \Delta I}{255} \left(\frac{1}{z_{near}} - \frac{1}{z_{far}}\right) + \frac{1}{z_{far}}$. The modification to the image is invisible when rendered normally when the value is larger than 1 for z_{near} . However, when the value is below zero, the effect becomes more visible and can be defined as $\Delta d = f \cdot t_x \cdot \frac{\Delta I}{255} \left(\frac{1}{z_{near}} - \frac{1}{z_{far}}\right)$. This watermark occurs by intentionally creating holes in the image, such that these holes normally take the form of a logo.

The goal of the watermarking scheme proposed in [31] is to embed a strong watermark into the image, while minimizing the visible effect in the image to reduce strain on the observer. This is almost identical to the goal of the watermarking scheme proposed in [28]. The author proposed that the reduction of the visible effect can be achieved through the identification of the Region of Interest (ROI). The ROI is the region where the user is most likely to concentrate because the foreground is of greater interest to the user than the background. The identification of the ROI depends upon combining the following three layers: (1) Thresholded Depth Map Layer, (2) Edge detection of Depth Map and (3) Edge detection of Center Image. Once the three layers are integrated, the combined image undergoes boundary scanning and generates a layer that identifies the region of interest. Then, the watermarking process divides the center image into 8 by 8 and embeds the information using the following formula: $X = S + \alpha(ROI) \cdot b \cdot w_p$, where the scalar value is high for blocks that are not within the ROI and low for block within the ROI.

4.3 Disparity Map Image Watermarking

Reference [32] proposes a stereoscopic image encoding scheme through the use of watermarking. This system begins by calculating the disparity map between the Left and Right Image for the Right Image, while creating a one-dimensional left vector using zig-zag sequence. This zig-zag sequence is rearranged into a 2D image creating a degraded left image. Afterward, the Fractional Fourier Transform (FrFT) is performed. To embed the watermark, the SVD (singular value decomposition) is used on the disparity map, yielding FrFT output. The SVD transform breaks an image into three separate matrices. The diagonal matrix from the decomposition of the disparity matrix is added to the diagonal matrix of the FrFT decomposition using the following formula: $\Sigma_{wm} = \Sigma_{LeftDegraded} + \alpha \Sigma_{Disparity}$. Using the diagonal matrix, Σ_{wm} and the two other matrices generated from FrFT decomposition, the process is reversed to produce a watermarked degraded left image. This is followed by reversing the 2D degradation of the zig-zag scanline, resulting in a watermarked left image. The watermark detection process is similar to the embedding process. However, the transforms are performed on the watermarked and non-watermarked left image. Thus we have $\Sigma_{wm} = \Sigma_{LeftDegraded} - \alpha \Sigma_{Disparity}$.

References [33] and [34] propose two different embedding locations for the watermark. The first location is in the right image prior to the disparity estimation, and the second location is in the disparity map after the disparity estimation. The system embedded the information using the Iprotect method. Iprotect is a hybrid method, which treats watermarking as an optimization problem. The system encodes and decodes the message. Based on the detected errors in the message, the pattern is modified and re-embedded. By iterating through the solution, the system will converge on an embedded signal that is capable of being recovered.

The work in [35] and [36] proposes a novel watermarking scheme while exploring the benefits of different frequency domain transformations. In addition, they evaluate different disparity map estimation techniques. This section will begin with a discussion of the embedding process used in [35] and [36] and continues by specifying the transforms and disparity map cost functions that were used. The embedding process begins through the transformation of the Right Image into a frequency domain, after which the values are quantized. For simplicity, assume the transform is one dimensional. This would yield a vector $RI = \{RI_0, RI_1, ..., RI_M\}$. Equivalently, the watermark

transform coefficients is $W = \{W_0, W_1, ..., W_N\}$, where N is smaller than M. The watermark coefficients are then reordered using a random seed. Afterward, the coefficients are used to replace some coefficients of the image, resulting in a new vector in the form of $RI = \{RI_0, RI_1, ..., T_0, T_1, ..., T_i, T_{i+1}, ..., T_N, ..., RI_M\}$, where $T_i = RI_i + \alpha |RI_i|W$. The reason that the middle coefficients are replaced is simple. If the lower frequency values are modified, the effect on the system would be too great. On the other hand, if the lower coefficients are replaced the watermark would be fragile to compression attacks. Afterward, the inverse transformation is executed. The disparity calculation is performed and the left image and the disparity map are transmitted. Using the random seed key, the watermark can be extracted and reordered to restore the initial embedding mark, but only after the right image is reconstructed, using the left image and the disparity map.

4.4 Limitations of Existing Work

Various watermarking schemes provide some advantages, but some limitations persist. For example, the system proposed in [24] requires both the left and the right image to arrive at the decoder, otherwise the content of the embedded information will be lost. This would allow a 2D version of the image to be illegally transmitted without being able to track the malicious user. This shortcoming also exists in the work presented in [25]. However, the work in [25] stored polar values in the image. While it is possible, in theory, to extract the watermark from a single image by using the original lost image, the authors did not address that possibility in the paper. This would introduce the need for more information to be stored at the detector to help extract the watermark.

The work in [27] presented a unique approach to embedding the watermark. The approach is capable of detecting the image from the left, right and center image. However, the largest limitation of the system is the low bit rate. This problem is shared by the work presented in [28], where larger watermark patterns are used. The work presented in [31] aims to limit the effects of watermark on the viewing experience and suffers from the same limitations as the system presented in [28], which are large watermark pattern and low bitrates. While [29] proposed embedding the watermark in all three versions of the image, the main drawback of this approach is its scalability such that the information that can be embedded within the image will decrease as the size of the image increases. This problem is caused by the design of the watermarking scheme where the images are rendered to embed the desired information. The second limitation with this watermarking scheme is its inability to

distinguish between watermarked and non-watermarked blocks. In contrast, the work in [30] aims to embed information within the watermark by distorting the image under specific rendering environments. This approach would be effective for the detection of copyright holder; however, it is incapable of embedding the information from a malicious user due to the unique properties of the watermarking scheme.

The work in [26] proposes storing information in the watermark on the right image. This watermarking scheme would make it impossible for the detector to retrieve any information from the left image and relies solely on the right image to be transmitted for the detection process. This problem is also faced in the work presented in [35], [37], and [36]. However, the work in [26] stored information in the disparity map, and this information can only be detected once the right image is rendered, while in [33] and [34] it is stored in the disparity map. Due to the disparity map being used as the coverwork, the information cannot be retrieved from the left, or the right image.

Chapter 5 Advanced Watermarking Concepts

This chapter will focus on addressing some challenges in detecting the watermarked pattern and the theory of the methods used to combat them. Section 5.1 will discuss the various types of attacks and metrics. Section 5.2 will focus on discussing watermark robustness and propose two methods for improving the performance of the watermarking system. Lastly, Section 5.3 will discuss some of the challenges faced in verifying the authenticity of the detected watermark and the recommended method for combating this problem.

5.1 Watermark Attacks and Metrics

The success of watermarking is based on its ability to survive attacks. Attacks are defined as any modification that can alter the image or the watermark pattern. These attacks can be deliberate, where a malicious user is attempting to remove embedded information or unintentional attacks due to image enhancement techniques or transmission errors. Thus, the watermark should be able to withstand deliberate attacks and photo enhancement. The work in [38] classified watermark attacks into four categories: (1) Removal Attack, (2) Geometric Attack, (3) Cryptographic Attack and (4) Protocol Attack.

Removal attacks focus on the removal of the watermark through image manipulations. These manipulations involve either (1) Filtering or (2) Noise Addition. The purpose of these attacks is to make detection of the watermark difficult without breaking the security of the watermark. Geometric attacks normally attempt to distort the watermark using rotation, scaling and translation attacks. Typically, these attacks can be combatted using synchronization counter measures. Cryptographic attacks attempt to crack security by removing the watermark or misleading the detector. Finally, protocol attacks attempt to add the attacker's watermark in order to create ambiguity.

The test in this thesis will focus on testing the system's robustness to removal attacks. Protocol attacks are not of great concern for reference tracking, because each user will gain a unique watermark pattern. If watermarked content does have more than one identifier it is safe to assume that both parties participated in the copyright infringement. On the other hand, while geometric attacks do affect the watermark their influence on the 3D experience is still unknown since most systems surveyed in Chapter 4 did not conduct such a study. Finally, a cryptographic attack test will not be conducted because these tests rely on brute force searches making them impractical, as discussed in [39].

To test the effectiveness of the system the following attacks were used: (1) Salt and Pepper Noise Addition, (2) Gaussian Noise Addition, (3) Median Filtering, (4) Blurring Filter, (5) Sharpening Filter, (6) Histogram Equalization, (7) JPEG Compression and (8) Cropping. A good percentage of intentional and unintentional attacks are covered through these eight methods, and insight will be gained into the robustness of the system. The details of these attacks will be discussed further in Chapter 7.

To help quantify the effectiveness of the watermarking systems, signal metrics must be used. The first and most popular metric is the Peak Signal-to-Noise-Ratio, which is generally used to measure the effect of watermarking on the system. This metric is defined by the following equation:

$$PSNR = 20 \cdot \log_{10} \frac{255}{\sqrt{MSE}}$$
(5.1)

Mean Square Error (MSE) is defined below:

$$MSE = \frac{1}{h \cdot w} \sum_{u=1}^{w} \sum_{v=1}^{h} \left(I_{orig}(u,v) - I_{mod}(u,v) \right)^2$$
(5.2)

where *h* is the height of the image, *w* is the width of the image, $I_{orig}(u, v)$ is the original image and $I_{mod}(u, v)$ is the modified image. The above equation applies only to gray scale images. If RGB images are used, then the MSE is the sum of all squared values divided by three times the number of pixels $(3 \cdot h \cdot w)$. The robustness of the extracted message must be measured. The work in [38] used the normal correlation to measure the difference between the embedded and extracted messages. Since the watermarking system in this thesis stored binary information, however, the Bit-Error-Rate (BER) will be used. The equation below defines the BER:

$$BER = 100\% \cdot \frac{E_{EM}}{S_M} \tag{5.3}$$

where, E_{EM} is the number of errors in the extracted message and S_M is the size of the message.

5.2 Watermark Robustness

One consideration for classifying watermarking schemes is their robustness. The robustness of a watermark is defined as its ability to remain intact when it undergoes signal distortion, as discussed in Section 5.1. Normally the robustness of a system is classified as either 1) Fragile: inability to survive distortion; 2) Semi-Fragile: ability to survive a limited amount of distortion; or 3) Robust: ability to survive most signal distortion. Creating a watermarking scheme that can survive all signal distortions with varying intensities is difficult. The intended application of this watermarking scheme is for transaction tracking, to ensure that malicious users (Example in 1.1) cannot remove the watermark; thus, robustness enhancement techniques must be used. To aid in this process two methods were used. The first is Error Correcting Code to help reduce the bit error rate. The second is Message Scrambling to defend against localized watermark attacks.

5.2.1 Convolutional Code

The various attacks mentioned in Section 5.1 attack the watermark pattern and cause the embedded information to be misinterpreted by the detector. To combat this issue, error correcting code can be used to correct the detected message. Error correction can be achieved through the addition of redundancy into the information, by passing it through an encoder. Normally, the encoder will provide several outputs, p_i . In this thesis convolutional codes were used.

Convolutional codes are classified based on three parameters: 1) rate, 2) constraint, and 3) polynomial. The Code rate (m/n) is usually a ratio between the provided number of bits (m) and the size of the output code (n). The constraint is the number of past and present inputs, which are used for the generation of the output. Finally, the polynomial describes the relationship between the inputs and the output nodes. The equation below provides the steps for generating the output:

$$p_i[n] = \sum_{j=0}^{k-1} g_i[j] x[n-j] mod 2$$
(5.4)

where p_i is the output bit, *i* is the number of outputs, *j* is the constraint, g_i is the generator polynomial that corresponds to the *i*th output, and *x* is the input bit stream. The generator polynomial

is the same length as the constraint and can hold a value of zero or one. An example of such a polynomial is the value [1,0,1]; when substituted into equation 5.4 the output equation becomes $p_i(n) = x[n-2] + x[n]$. For simplicity, all diagrams and examples in the rest of this section will use a 1/2 rate code, with a constraint size of 3, where $g_1 = [1,1,1]$ and $g_2 = [1,1,0]$.

Two main representations for convolutional codes are illustrated in Figure 5.1. Figure 5.1 a, is the block diagram representation of the encoder, while 5.1 b is a state machine representation of the convolutional code. While there are several generator polynomials, not all polynomials have good error correcting properties. Reference [40] explains the process required for measuring the effectiveness of a generator polynomial and provides a table of the more effective encoding polynomials.



Figure 5.1: (a) Block Diagram of Convolutional Code, (b) State Machine Representation of Convolutional Code

The most basic method of decoding convolutional codes is to generate all possible combinations and compare the distance between the generated output and the received signal. The distance is usually evaluated using the Hamming distance, from [40], which is the disparity between two sequences. This value usually indicates how many bits need to be substituted in order to change one sequence into the appearance of the other. However, the primary disadvantage of this approach is that as the complexity of the code increases, the time needed to decode the sequence increases--for every extra bit, the number of comparisons double. To combat this problem Viterbi decoders are used [40]. The Viterbi decoder's main goal is to explore all branches within the finite state machine. The algorithm treats each state as a node; at every node the Viterbi decoder branches into two new paths. As the decoder creates more paths, it evaluates the Hamming distance of the path. Occasionally two paths converge onto the same node. Once this occurs the decoder will chose the path with the lowest Hamming distance and will delete the other. It continues branching the paths until the size of the paths are the same as the received message. Once again the decoder compares the Hamming distance of all paths and chooses the path with the lowest Hamming distance and returns it as the output. The advantage of this approach is that, unlike the brute force approach, the decoder deletes paths as it becomes clear that they will not results in a correct output. Figure 5.2 illustrates the path generation mechanism of the Viterbi decoder path exploration. For simplicity, the path sizes were limited to three levels. The diagram presented below is often referred to as the trellis.



Figure 5.2: Viterbi Decoder Branching

5.2.2 Scrambling

Scrambling is generally used to increase the watermark's robustness against localized attacks. The scrambling can be applied in either of two locations: (1) Image or (2) Message. While image scrambling is more robust against cropping attacks, it is not feasible for the watermarking system used in this thesis. Normally, scrambling is a pre-processing step that is applied prior to embedding the watermark. This scrambles the pixels across the image plane. Thus, if a localized attacked, such as a cropping attack is ever applied, the damaged pixels will not significantly affect watermark detection. While the entire block is damaged, the scrambling spreads pixels across multiple blocks, thus reducing the effects on the overall system. Even though this is the most robust approach, it is also not feasible for the system outlined here because the essential pixel locations would be marred by the shifting caused by the image rendering process. However, to maintain some of the benefits of

scrambling, the bits of the message are scrambled. This approach ensures that localized attacks do not damage related bits. It also ensures that the Hamming distance is not significantly influenced by localized attacks, ensuring that the path reduction mechanism within the decoder is not negatively influenced.

5.3 Message Authentication

1	Let <i>B</i> represent input bit message				
2	Let C_S represent size of CRC				
3	Let <i>P</i> , represent the CRC Polynomial				
4	$B_A = B$ appended with C_S zeroes				
5	for $i = 1$ to Length of <i>B</i>				
6	if $(B_A[i]$ equals 0)				
7	Do Nothing				
8	else				
9	for $k = 1$ to C_S				
10	$B_A[i+k-1] = (B_A[i+k-1] + P[i+k-1]) \mod 2$				
11	end for				
12	end if				
13	end for				
14	return the Last C_S bit of B_A				

Algorithm 5.1: CRC Code Generation Algorithm

The watermark detection process usually places a threshold (Equation 3.3) on the correlation to ensure that no false detections are made. Nevertheless, the possibility of false detection remains. A major challenge in watermark detection is verifying that the extracted message is correct. If a watermark decoder was to be used on a non-watermarked image, there would be no way of authenticating the validity of the extracted message. To combat this problem Cyclical Redundancy Codes (CRC) were used in this thesis, similar to the work in [31]. The benefit of such a system is that

it significantly reduces the likelihood of false detection and is capable of detecting errors within the message.



Figure 5.3: CRC Code Generation

CRC codes are error detecting codes in this thesis. The single wire implementation of CRC codes was used. This code is capable of detecting the following errors: (1) Any odd number of bit errors within the message, (2) All double bit errors within the message, (3) Consecutive errors within a windows that are equal to or smaller than the CRC size, and (4) Large Clusters of Errors. To calculate the CRC, a polynomial divisor is used. There are two ways to represent the bit divisor, the first of which is as a bit string, 1001. The second way is as a polynomial , $x^4 + 1$. Figure 5.3 presents the method for calculating the CRC value of a Message, using $x^8 + x^5 + x^4 + 1$ as the polynomial, where a value of 0 is stored in the flip-flop registers during the initialization process. Afterward, the input bits are sent into the system one bit at a time. Upon completion, the final value in the registers represents the CRC value. A second method of calculating the CRC using software is presented in Algorithm 5.1; this process is illustrated in Figure 5.4.



Figure 5.4: CRC Generation Flowchart

Chapter 6 Proposed System

This section will present the realization of the proposed system. The objective of this chapter is to expand upon the theory presented in the previous chapters and present a watermarking system for 3D disparity-based images. Section 6.1 will provide an overview of the proposed system's building blocks. Section 6.2 will focus on all the necessary steps that are performed prior to the embedding process. Section 6.3 will discuss the watermarking process and how the information from Section 6.2 will be used. Finally, Section 6.4 will discuss the detection process itself.

6.1 System Overview



Figure 6.1: The proposed System Overview

For simplicity, the overall system is broken into three separate layers. The first is the embedding layer, which is responsible for embedding the watermark within the left image. Next is the transmission layer, which contains the disparity map and the watermarked left image. The final layer is the detection layer. The embedding layer is further broken down into the key components of (1) Pre-processing and (2) Embedding. Prior to the embedding process, the disparity map of the images must be calculated as well as the total distortion that will occur from the rendering process. This information is used to generate a decision matrix that will be used by the embedder. To help increase the robustness of the system, some message pre-processing is used. This pre-processing helps

increase the message's bit recovery rate. After these two tasks are performed, the embedding process takes the information provided and embeds the message into the image. Upon the completion of the embedding process, the information is sent across the transmission layer. Once the information arrives at its intended destination, the content is used. If the protected content is redistributed on unauthorized channels, the detector is used to identify the malicious user. This process is presented in Figure 1.1.

6.2 Pre-processing

6.2.1 Disparity Map



Figure 6.2: Disparity Map Estimation Process

The first step in this process is the generation of the disparity map. The disparity map uses the minimization process discussed in Section 2.3 to create a disparity map of the image. Note that the values of the disparity map vary, and the search range needs to be limited to 3% of the image width. Upon the completion of the disparity map estimation process, the disparity map undergoes some

filtering. The filtering process allows the disparity map to be smoother, creating a better rendered image. The added advantage is that it increases the disparity map's resistance to filtering attacks and aids the system's ability to survive attacks.

The main challenge in the disparity map process described thus far is that the cropping of the image is removed. Due to the nature of the disparity maps, certain regions of the image cannot undergo disparity estimation. To combat this problem, the images are zero padded where the size of the padding is equal to the windowing function. This step helps eliminate the cropping on the edges of the image. After the completion of this step, the padding is removed. To eliminate the limit of the search range, two disparity maps are generated. The first map generated is achieved by treating the left image as the reference, while the second map treats the right image as the reference. The two maps are, then, combined based on a "winner-take-all" principle. Thus, the disparity value in the left or right disparity map with the lowest cost function value is treated as the true disparity, as demonstrated in Figure 6.2.



6.2.2 Hidden Pixel Disparity map

Figure 6.3: Hidden Pixel Example

The decision matrix is used to predict the holes and hidden pixels within the image. An example of hidden pixel is provided in Figure 6.3. This occurs when two pixels occupy the same space location in the rendered image. One of these pixels will be lost in the rendered image. In the context of watermarking, this information loss will cause a loss of potency in the embedded watermark.



Algorithm 6.1: Hidden Pixel Disparity Map Generation Algorithm

A prediction procedure is used to help predict where these events will occur. At first, a faux-left image is generated. This faux-image has the same dimensions as the original image, whose values increase along the x-axis. Normally, pixel values are limited to a maximum of 255 per pixel, but this limit is removed in the faux-image to ensure pixel uniqueness. These images are defined as $I_{GL}(x, y)$ for a faux-left view and $I_{GR}(x, y)$ for a faux-right view. This image is sent into the image renderer along with the generated disparity map. This creates two faux-images, a left and a right image, which undergo the procedure presented in the algorithm. This algorithm is a modified version of the disparity map estimation process; because pixel uniqueness is ensured, a cost function is unnecessary. The result of this algorithm is a merged disparity-map and hidden pixel map, presented in Figure 6.6. The algorithm is presented above (6.1). This process is illustrated in the figure below (6.4).



Figure 6.4: Hidden Pixel Disparity Map Generation Flowchart

6.2.3 Message Processing



Figure 6.5: Embedded Message Format

The message formatting plays an extremely important role in the system. Due to the nature of the proposed scheme, discussed in Section 6.2.2, the storage capacity varies from image to image. However, this value is usually determined by the number of available blocks. Let N_A be the number of available embedding locations. The total number of bits that can be stored is defined by the following equation:

$$S_M = (N_A - P) \cdot k - C_S \tag{6.1}$$

where S_M is the total number of available bits, k is the convolutional code rate, C_S is the size of the CRC code and P is the polarity header. The value of available bits varies from image to image due to the volatility of the decision matrix. The structure of the message is presented in Figure 6.5. From the structure it is clear that the first P bits of the system play no role in the detection process. However, some watermark attacks invert the polarity of the embedded message. To help combat that problem, the polarity header is placed at the beginning of the message. In this way a reference mark is placed to help identify the polarity of the embedded message. The second portion of the message is the convolutional encoded message. This bit sequence is composed of the desired message and the CRC of the embedded message.

6.3 Embedding Process

6.3.1 Decision Matrix



Figure 6.6: (a) Hidden Pixel Disparity Map, (b) Decision Matrix

To embed messages of multiple bits, the image is split into square blocks of size 8x8. Each of these blocks contains a single bit. Prior to the embedding process, the information of the hidden pixel disparity map is used to reach a decision about which regions should undergo embedding and which ones should be omitted. The Hidden Pixel Disparity Map (HPDM) is broken into blocks. Each of these blocks is checked to ensure that it contains no holes and that all disparity values within this block are equal. This ensures that no hidden pixels exist, while eliminating other forms of disparity map errors. Figure 6.6 presents an example of the generated decision matrix and the HPDM. This matrix should be stored because it will play a significant role in the detection process of the two images.

6.3.2 Embedding Process



Figure 6.7: Embedding Process

The embedding approach follows the process presented in Figure 6.7. At first, the image is broken into equal-sized blocks. Each of these block undergoes quantization (using the matrix discussed in Section 3.4.3) followed by a frequency transform using the DCT. These values, then, undergo the zig-zag scan-line transformation. Upon the zig-zag transformation, the middle frequency bands are selected and the watermark pattern (w_m) is added using the equation presented in Section 3.2.1. Upon the addition of the watermark pattern, the mid-frequency vector is reinserted into the zig-zag scan-line of the image. This process is reversed along with the quantization and the DCT transform, resulting in a watermarked block.
6.4 Detection Process



6.4.1 Watermark Extraction Process



The watermark extraction process breaks the received image into blocks; once again it converts the frequency matrix into a vector and then selects the middle frequency bands. Using the correlation coefficient formula, along with the threshold equation described in Section 3.2.2, a decision is made about the embedded bit. The decision matrix is used to determine which blocks contain an embedded bit. Once the information from all blocks is extracted, the first *P* bits (polarity header) are extracted if they follow the predefined format, and then the mechanism moves on to the next stage of the watermarking process. Otherwise, the polarity of the extracted bits is inverted. The remaining bits are decoded using a Viterbi decoder upon the completion of the decoding process. The last C_S bits are removed from the message. The resulting message is sent into a CRC block. If the output of this block matches the removed bits, the extraction process has succeeded. This process is illustrated in Figure 6.8.

6.4.2 3D Image Decoder

Due to the nature of 3D rendering, the detection process is slightly more complex. Figure 6.1 illustrated the detection process as a whole. Once the detector receives an image, it is not capable of detecting whether it is the left or right view. For this reason, the system has two watermark extraction paths. The first path assumes that the received image is the left image, while the second path assumes the received image is the right one. The primary difference between the two paths is that the right path reverses the rendering process using the decision matrix. Afterward, the watermark extraction

process described above is followed. If one of the paths declares that the extraction process has succeeded, then that message is sent to the user. Otherwise, the decoder provides no message.

6.5 Practical Realization of Rendering

.

.

While the rendering process described in Section 2.4 is the theoretical approach definition, it suffers from practical realizations. To combat this problem the proposed system modifies the rendering process to better reflect the nature of the disparity that occurs between the two image pixels.

This is achieved by breaking the rendering process into $(d_{max} + 1)$ (the number of disparity values) steps. Each of these steps requires that a mask be created, where m_i is the mask. Each of these masks corresponds to a specific disparity value; for example, mask m_1 correspond to all pixels within the image that will incur a horizontal shift of 1 (disparity map value of 1).

1Let
$$D(k, j)$$
 be the disparity map2Let I_R be the rendered image3for $m = 0 \dots d_{max}$ 4for $j = 1 \dots y_{max}$ 5for $k = 1 \dots x_{max}$ 6if $D(k, j)$ equals m 7 $I_R(k, j - D(k, j)) = I_L(k, j)$ 8end if9end for10end for10end for

Algorithm 6.2: Realistic Rendering

After creating the masks, the system shifts the pixels of the image one layer at a time, beginning with mask m_0 , and keeps incrementing until mask $m_{d_{max}}$, using Equation 2.21. This approach breaks the condition placed in Equation 2.22. It is a far more practical realization of the

system for the following reasons: (1) the rendering is more realistic, and (2) the direction of the rendering does not affect the output image. To elaborate on the first benefit, it is far more likely that a foreground pixel (higher disparity value) will occupy a pixel location than it is for a background pixel (lower disparity value) to do so. By rendering the background first and allowing the foreground to replace the pixel, far more realistic scenes are created. With regards to the second reason, assume that two adjacent pixels occupy the same pixel location in the rendered image. If the rendering process were to begin by rendering from the left side of the reference image, the left-most pixel would occupy the space in the rendered image. If the rendering process were to begin from the opposite direction (right), the right-most pixel would appear in the rendered image, when subject to condition 2.22. Algorithm 6.3 describes this process, and it is illustrated in the figure below (6.9).



Figure 6.9: Realistic Rendering Flowchart

6.6 Assumptions and Observations

Throughout this thesis, there were a few assumptions made in the design of the system. Many of these affected the design decision made for the proposed system presented in Sections 6.1 - 6.5. This section begins by discussing some of these assumptions, which influenced the objectives of the presented work. Special observations are discussed afterwards.

6.6.1 Assumptions

The demand for digital distribution of media content is increasing drastically. This increase in distribution will, in turn, proliferate to 3D content, which lacks the necessary copyright protection mechanisms that 2D content currently enjoys. To resolve this problem, the work in this thesis focuses on the use of watermarking for the purpose of copyright protection.

While a watermarking system (Figure 1.1.) cannot prevent copyright infringement, it can identify the users responsible for the redistribute content on unofficial channels. To be effective, the watermarking system needs to embed an identifier for each active user on the distribution service. This approach assumes that each person is allowed a unique identification number. As a safety precaution, a multiplier of five was chosen to account for population growth and multiple accounts. While this is an excessive value, it provides large overhead ensuring that no identification number collision will occur for the foreseeable future. Of the existing watermarking systems, block based systems are the only ones capable of maintaining high robustness with a data payload capable of meeting the requirements.

Existing 3D content uses stereoscopic image pairs (Figure 1.7 (a)). While widely popular, this format suffers from a significantly larger data requirement. To reduce this requirement, the disparity map provides the best compression ratio (Figure 1.7 (c)). Therefore, the proposed watermarking scheme adopted disparity map estimation. The added benefit of such a system is the ability to convert depth map-based images into disparity map-based images, as demonstrated in [12]. This conversion would allow the depth map-based systems to benefit from smaller data sets than disparity maps.

Adapting existing content ensures that the two provided images are already rectified, as described in Section 2.2. This permits the disparity map optimization to be reduced to allow search in the horizontal axis. This approach provides two primary benefits: (1) computational complexity of disparity map is significantly reduced and (2) disparity map required for transmission contains half of the information necessary to reconstruct the right image, reducing its transmission size.

6.6.2 Design Observations

The primary objective of the proposed system is to be usable in a transaction tracking capacity. To meet this goal, the proposed system must be robust to various attacks. For this reason, the DCT transformation was used. Compared to other watermarking schemes, DCTs suffer from higher computational complexity. However, such complexity allows the system to use spread spectrum techniques, resulting in an increase of the survival rate for the embedded message and reducing the malicious users' ability to evade detection.

The added benefit of the DCT transformation is the ability to combat JPEG compression when combined with the quantization matrix, as discussed in Section 3.4.2. This is attributed to the proposed system's ability to better combat some of the losses faced when images undergo JPEG compression. The selection of DCT played a key role in the decision to use 8 by 8 block sizes, with the goal of matching the size of the quantization matrix.

As for the detection process, it was clear that due to the use of spread spectrum techniques, a correlation-based detector would provide the best detection. Primarily three correlation functions can be used, as follows: LC, NC and CC. However, the large detection region of the LC function (Section 3.2.2) results in false detection. The aforementioned problem is solved in NC; however, NC suffers from poor detection when the image is scaled due to standard image enhancement techniques, such as biasing and histogram equalization. Therefore, the CC function is the only correlation capable of meeting the requirements.

To further improve the robustness of the system, scrambling was used. The two primary benefits for such a system are (1) security and (2) robustness against localized attacks. The added security results from randomly scrambling the message. Since the data is randomized, a malicious user will have an extremely difficult time detecting the message and inserting false information. The increased robustness, however, can only be achieved when combined with convolutional codes. Spreading the information to random regions across the image prevents dependent bits from occupying adjacent locations. This dispersion ensures that a potential localized attack will have a minimal effect on the overall message, because the damaged bits would be spread out across the convolutional code.

Another issue in the detection process is that some attacks reverse the polarity of the embedded pattern. Polarity headers are used to solve this problem. A large header results in a higher success rate. For this project a header of size seven was chosen. The polarity of the detected bits can be determined by comparing the detected header with the embedded header (using Hamming Distance metric); therefore, polarity reversal can be prevented.

The final issue in the proposed system is its inability to distinguish between left and right images. To solve this problem, there are two detection paths within the watermark detector. The first path assumes the image is a left image, while the second path assumes it is a right image. After decoding the embedded message, the detector compares the CRC of the detected message to the CRC of the embedded message. If the two messages match, the system suggests that the message has been detected properly.

The reason that the system cannot distinguish between the two formats is that the images do not have a unique identifier. The system could compare the images to the original left image and make a decision based on the image similarity. However, the aforementioned process would increase the amount of information that the detector must store. Therefore, the sixteen-bit loss in the data payload is a smaller penalty.

6.7 Comparison with Other Work

6.7.1 Watermark Characteristics

To evaluate the proposed system, it is important to compare its characteristics with similar 3D image watermarking systems. Table 6.1 presents a comparison between the proposed system and the works discussed in the literature survey in Chapter 4. The blue column corresponds to the proposed system. A green cell indicates an improvement in the proposed system, a yellow cell indicates no improvement, and a red cell indicates the proposed system behaves negatively under these circumstances. Some of the characteristics provided in the data are quantitative, but a few are quantitative. The first three rows, which correspond to the watermark characteristics, are qualitative because they discuss the embedding location. While the following three rows (Decoding Requirements) are quantitative because they refer to the storage requirements necessary to decode the watermark from the left image. Furthermore, the data scalability is a qualitative property determined by evaluating the structure of the proposed system. Finally, the payload characteristic is both a

qualitative and a quantitative property. The qualitative aspect of this metric refers to the reliability of available bits from image to image. On the other hand, a quantitative element of the property refers to the average number of bits that can be embedded.

To meet the transaction-tracking objective, the system must be capable of recovering information from the left and right images. Thus the watermark location is its most significant characteristic. From the table, it is clear that the proposed system outperforms the majority of the previous systems. Exceptions are the systems proposed in [27], [29] and [30]. However, there are a couple of drawbacks: (1) decoding data requirement and (2) data payload.

The decoding data requirement indicates the amount of data needed for extracting information from the received image. In the data payload category, it can be observed that relative to the work performed in [24], [32], [33], [34], [35] and [36], the proposed system does not outperform the reported work. However, when compared to the above reported works, excluding [24], the proposed system outperforms these systems in the area of decoding data requirements.

Compared with the works in [27], [28] and [29], there was no benefit in the proposed watermarking scheme regarding the watermark location, while a compromise was made for the decoding data requirement metric. However, when compared to the work presented in [27] and [30], the proposed system could store more information (data payload characteristic), meeting one of the primary objectives of the proposed system. Finally, when compared to the work in [29], the system was capable of higher scalability. Scalability is defined as the system's ability to increase its data payload consistently as the resolution of the provided image increases. The improvement in scalability is also observed when the proposed system is compared with the work in [28]. While the key required category is a significant component of the detection process, its low data overhead (4-5 bytes) would not create a large difference in the data so it was not considered in the decoding requirement comparison.

In conclusion, while the proposed system has some limitations, it outperforms the work presented in the literature in the categories necessary to effectively prevent malicious users from manipulating the transaction tracking system.

		Disparity Map	Stereo Image			Depth Map					Disparity Map		
		Proposed System	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33] and [34]	[35] and [36]
ation	Right Image Embedded	Yes	Partially	Partially	Yes	Yes	Not Clear	Yes	Yes	Not Clear	No	Yes	Yes
nark Loc	Left Image Embedded	Yes	Partially	Partially	No	Yes	Not Clear	Yes	Yes	Not Clear	Yes	No	No
Watern	Center Image Embedded	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Yes	Yes	Yes	Yes	Yes	Not Applicable	Not Applicable	Not Applicable
nts	Keys Required for extraction	Yes (One)	Yes (One)	Yes (two)	Yes (One)	Yes (Two)	Yes (One)	Yes (One)	Potentially (One)	Yes (One)	Yes (One)	Yes(One)	Yes (One)
Requireme	Decision Matrix Required For Extraction	Yes	Yes	No	No	No	No	No	No	No	No	No	No
Decoding	Original Image Required	No	No	No	Disparity Map	No	No	No	Generated Depth Map	No	Left Image	Generate Disparity Map	Generate Disparity Map
	Data Scalable	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
Data	Data Payload Characteristics	Variable (Image Structure)	Constant	Variable (Global Disparity)	Variable (Image Structure)	Constant (Low)	Constant (Low)	Variable (Image Structure)	No Data Payload	Variable (Image Structure)	Constant	Constant	Constant

Table 6.1: Comparison of System Characteristics

6.7.2 Novelty of the Proposed System

When designing the proposed system, there were various influences from the existing literature that inspired the different components that were used. The primary components inspired by external sources were: (1) disparity map preprocessing, (2) HPDM and (3) watermark processes.

The initial inspiration for the pre-processing of the disparity map came from the work in [35] and [36]. The motivation behind the adoption of this mechanism is the lack of smoothness within the disparity map. When the disparity map (Figure 1.7 (c)) is compared to the depth map (Figure 1.7 (b)), the lack of uniformity of the disparity map becomes clear. To resolve this problem, the works from [18] and [19] were used to help reduce the disparity map's noise.

The design of the most significant component of the system, i.e., HPDM, was influenced by the work performed in [28] and [29]. The work in [28] implemented a system that embeds the information into all available pixels. To help reduce some of the distortion that occurs, the work located the pixels that would be hidden and decreased the embedding strengths at those locations. However, this approach is capable of embedding one bit per image, while the work in [29] renders a block of an image and embeds information if a certain criterion is met. This approach is not scalable as the image resolution increases with the value of the disparity map. This change will cause the system to have fewer available blocks. The process is presented in Section 2.3.

In this thesis the rendering process was expanded to the entire image. The hidden pixels were then detected from the rendered frame. Using the non-hidden pixels from the rendered image, a new disparity map was created. This new disparity map, called the Hidden-Pixel Disparity Map, is capable of detecting the holes and blockings within the image. This information can be used to prevent the encoder from embedding information within those blocks. However, to detect the embedded message, a reduced form of the HPDM, i.e., decision matrix, must be stored for the retrieval of the image. This process is described in detail in Section 6.2.2.

The two primary components for the watermarking process are (1) encoder and (2) decoder. These components drew inspiration from various sources. The components inspired by external sources were: (1) spread spectrum addition, (2) quantization, (3) pattern generation and (4) correlation. The spread spectrum technique was inspired by the work presented in [22]. However, the approach presented in the paper had to be altered in this thesis, because the method in [22] is capable of transmitting one bit per image. The Quantization technique is rather popular, and it was inspired by the work in [35] and [36]. The primary difference is that while the works presented in [35] and [36] do not reverse the quantization process, the work presented in this thesis does.

As for the pattern generation, a better spread spectrum code was needed. Therefore, the work in [23] was used to guide the selection of the most appropriate PNG generator. The final component inspired by an external source was the correlation block. It was influenced by the work performed in [20].

The majority of the influenced components mentioned above can be found throughout Chapter 6, as listed below:

- Disparity Map Preprocessing, red sub-block in Figure 6.1.
- HPDM, blue sub-block in Figure 6.1.
- Spread Spectrum Addition, red sub-block in Figure 6.7
- Quantization, blue sub-block in Figure 6.7.
- Pattern Generation, green sub-block in Figures 6.7 and 6.8.
- Correlation, purple sub-block in Figure 6.8.

Chapter 7 Results and Evaluation

The results of this thesis are applied to images from the Middlebury data set [17]. The Middlebury data set also includes the Tsukuba data set, the results of which were used to create the various figures in this thesis. The first section will tabulate the resulting bit capacity of the various images (Tsukuba, Sawtooth, Venus, Bull, Poster, Barn1, and Barn2). The second section will provide the PSNR value of the images under various embedding strengths. Finally, Section 3.2 will test the robustness of the system under various attack conditions.

Bit Capacity	SSD	ZSSD	LS	SGRAD	SAD	ZSAD	LS	NCC	ZNCC	MSE	Theoretical
			SSD				SAD				Limit
			550				JAL				Linne
Tsukuba	799	850	859	981	665	835	831	849	778	850	1728
Sawtooth	2063	2057	2055	2108	2087	2093	2092	2051	2056	2057	2538
Venus	2052	2054	2049	2072	1993	2052	2050	2064	2077	2054	2538
Bull	2188	2217	2213	2192	2155	2208	2213	2221	2207	2217	2538
Poster	1926	1944	1991	1976	1879	1964	1962	2030	2024	1994	2538
Barn1	2024	2110	2123	2142	1976	2105	2117	2127	2113	2110	2538
Barn2	2189	2237	2237	2171	2164	2227	2223	2234	2201	2237	2538

7.1 Bit Capacity

 Table 7.1: Bit Capacity of the Image

The first thing that should be evaluated is the bit capacity of each image for the proposed system and how these results vary with different cost functions. These results are presented in Table 7.1. From the results of the bit analysis, it is clear that the method used in this thesis for generating the disparity map plays a significant role in the system's bit capacity. From the results it is also evident that for the majority of the images that used the SGRAD, they also had the best bit storage performance. For that reason, this disparity map method will be used for the remainder of the tests.

Another significant phenomenon that can be observed is the high fluctuations in bit efficient percentage (the number of embeddable bits divided by the theoretical limit). It can be especially observed in the Tsukuba data set due to the complexity of the scene. For example, if we were to divide the scene into two depth levels (for simplicity) -- the foreground layer and the background layer--when the rendering process occurs, objects in the foreground will overlap and cover objects in the background, increasing the number of hidden pixels. In turn, it will reduce the number of locations where the pixels can be embedded, as illustrated in Figure 7.1. As the number of objects in the foreground increases, the number of hidden pixels also increases.

This problem is further exacerbated by an increase in the number of disparity layers that exist within the image. While the number of embedding locations can be decreased by reducing the number of conditions set when generating the decision matrix, it would require a larger embedding strength and significantly reduce the fidelity of the image.



Figure 7.1: (a) Original Two Layer Image, (b) Rendered Two Layer Image

7.2 Fidelity

To measure the effect that the proposed watermarking scheme has on the image a quantitative metric must be used, in this thesis the average PSNR. This value was calculated by embedding a stream of 0 bits or a stream of 1 bits. The average for both images (Image embedded with 1, and Image embedded with zero) is taken and presented in Table 7.2. From the table it is clear that after an embedding strength of 0.5, the average drop in PSNR begins to approach an asymptote. Ideally, the higher the

Embedding Strength (α)	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Tsukuba	45.64	40.68	37.34	36.34	35.89	35.60	35.38	35.17	35.04	34.96	34.86
Sawtooth	45.69	39.49	35.89	34.85	34.38	34.09	33.86	33.64	33.51	33.42	33.32
Venus	45.66	39.59	35.99	34.95	34.48	34.19	33.96	33.74	33.61	33.52	33.42
Bull	45.66	39.37	35.74	34.69	34.21	33.92	33.69	33.47	33.34	33.25	33.15
Poster	45.73	39.76	36.19	35.16	34.70	34.41	34.18	33.96	33.83	33.74	33.63
Barn1	45.65	39.43	35.82	34.78	34.32	34.02	33.79	33.57	33.44	33.35	33.25
Barn2	45.72	39.40	35.76	34.70	34.22	33.93	33.70	33.48	33.35	33.26	33.16

PSNR is, the greater the image quality would be. For this reason an embedding strength of 0.25 will be used in the tests performed in Section 7.3.

Table 7.2: PSNR Values

7.3 Robustness

7.3.1 Noise Attack

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(\frac{x}{\sqrt{2\sigma}}\right)^2}$$
(7.1)

The first noise attack uses additive white noise to alter the received image. To simulate this attack a noise layer was generated and added to each color channel, where each color channel received a unique noise layer. For these attacks the only parameter that was altered was the standard deviation, σ , while the mean was maintained at zero. The statistical property of the noise abides by Equation 7.1. The results for this attack are presented in Table 7.3. This attack was not performed on the disparity map, due to the high level of distortion. This high level of distortion caused the rendered right image to be unusable. Clearly, there is no need to perform this attack because there would be no benefit for the attacker.

Standard Deviation (σ)	3	5	8	10	13	15	18	20	23	26
Tsukuba	0	0	0.88	1.3	2.22	4.44	6.22	7.55	7.11	11.11
Sawtooth	0	0.62	1.03	2.06	4.95	4.95	6.61	8.26	10.95	11.98
Venus	0	0	0.62	3.1	3.51	5.8	6.61	7.85	11.98	16.94
Bull	0	0	0.94	1.13	4.53	5.29	5.29	8.88	8.51	14.17
Poster	0.23	0.45	0.68	2.04	4.98	4.08	9.75	9.75	11.11	8.84
Barn1	0	0	1.23	2.48	4.75	7.02	7.85	7.64	13.02	13.85
Barn2	0	0	1.13	2.08	2.27	4.73	7.18	8.88	12.47	11.15

 Table 7.3: Gaussian Attack Results

From Table 7.3 it is clear that the system can withstand Gaussian attacks with a standard deviation of 5. While there were some outliers, this performance could be further enhanced by increasing the strength of the embedding process. However, the decrease in image fidelity would be significant. The average PSNR from this attack was 30.6 dB for a standard deviation of 5. The change in the PSNR did not change significantly as the attack strength was increased.

Pepper and Salt (α)	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Tsukuba (Left)	0	0	0	0	0	0	0.89	0.89	1.78	3.11
Sawtooth (Left)	0	0	0	0	0.21	0.21	0.62	0.62	2.07	2.89
Venus (Left)	0	0	0	0	0	0	0.83	0.21	1.24	0.62
Bull (Left)	0	0	0	0	0	0.19	0.19	0.19	0.19	0.57
Poster (Left)	0	0	0.45	0	0	0	0.23	0.91	1.36	1.81
Barn1 (Left)	0	0	0	0	0	0.62	0	0.83	1.24	2.69
Barn2 (Left)	0	0	0	0	0	0	0	0.38	0.95	1.7

Table 7.4: Salt and Pepper Results Left Image

$$p(z) = \begin{cases} \frac{\alpha}{2} & \text{if } z = a \\ \frac{\alpha}{2} & \text{if } z = b \\ 0 & otherwise \end{cases}$$
(7.2)

The second noise attack used salt and pepper additive noise. Similar to the Gaussian noise attack, a noise layer was generated and added to each color channel. Equation 7.2 defines the statistics of the added noise, where α is the probability of salt and pepper noise occurring. The results are

presented in Table 7.4. From the table it is clear that the watermarking scheme can withstand salt and pepper noise, with a 5% chance of the noise occurring. The PSNR of the attacks ranged from 47 dB, for 1% attack, to 37 dB, for 10%. The salt and pepper noise attack was also tested on the disparity map, but the influence on the generated right image was minimal, unlike the Gaussian noise attack. The results presented in Table 7.5 indicate the difference in BER between the right image and the left image. From the results it is clear that the system can withstand attacks of up to 3%. The visual quality of the image began to degrade significantly after the 5% threshold.

Pepper and Salt (α)	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
Tsukuba (Disp)	0	0	0	0.44	0.44	0	0.44	0	0	1.78
Sawtooth (Disp)	0	0.21	0.21	0.41	0.21	0.83	1.03	1.44	2.06	2.89
Venus (Disp)	0	0	0	0.21	0.41	0.41	0	0.83	1.45	3.72
Bull (Disp)	0	0	0	0.19	0	0.19	0.19	0.57	0.95	1.51
Poster (Disp)	0	0	0.23	0.45	1.13	1.36	1.81	2.72	3.17	4.76
Barn1 (Disp)	0	0	0.41	0.21	1.24	1.86	1.65	2.48	3.1	4.13
Barn2 (Disp)	0	0	0	0	0.19	0.76	1.9	2.65	4.35	4.35

Table 7.5: Salt and Pepper Results Right Image

7.3.2 Filtering Attacks

To test system robustness again, some image modification techniques were used. Three filter attacks were performed. Median filtering was the first since it is a popular technique for removing additive impulsive noise. The second filtering attack is known as average filtering. It is a popular form of noise removal that is effective at removing Gaussian white noise. The third filtering attack used was the sharpening filter. From the results presented in Table 7.6, it is clear that the watermarking scheme is more robust towards filtering attacks with larger mark sizes. Even though these results varied from image to image, it is clear that they could be improved by increasing the size of the embedded mark or the embedding strength of the watermark pattern. The majority of the presented attacks yielded an image with an average PSNR of 20 dB.

Filter Attack	Median 3x3	Median 5x5	Blurring 3x3	Blurring 5x5	Sharpening 3x3	Sharpening 5x5
Tsukuba	0.44	0.89	0	0	0	0
Sawtooth	11.36	1.03	5.37	0	0	0
Venus	9.92	2.27	7.43	0	0	0
Bull	6.23	0	0	0	0	0
Poster	24.94	7.93	29.25	0.22	0	0
Barn1	21.69	2.47	23.55	0	0	0
Barn2	17.20	0	2.08	0	0	0

Table 7.6: Filter Attacks

7.3.3 Miscellaneous Attacks

The final tests that need to be performed do not belong to the previous two categories, but they will be presented in this section. The first attack is Histogram Equalization. As expected, it did not cause any errors in the detected message. The second attack, image cropping, removed about 10 percent of the pixels in the image from one of the four sides of the image and attempted to detect the image, thanks to the use of message scrambling and Viterbi decoding. Finally, the watermarked image underwent JPEG compression to test the system's robustness against compression attacks. The results of these attacks are presented in Table 7.7. From this table it is clear that the system is capable of withstanding cropping and histogram attacks, while being able to withstand most JPEG compression. This indicates that the proposed system is robust and capable of withstanding most of these attacks.

Misc. Attacks	Histogram	Crop	Crop	Crop	Crop	Jpeg	Jpeg	Jpeg
	Equalization	Upper	Lower	Left	Right	40	60	80
Tsukuba (Left)	0	0	0	0	0	13.33	0	0
Sawtooth (Left)	0	0	0	0	0	3.71	0	0
Venus (Left)	0	0	0	0	0	22.93	0	0
Bull (Left)	0	0	0	0	0	20.41	0	0
Poster (Left)	0	0	0	0	0	1.85	0	0
Barn1 (Left)	0	0	0	0	0	8.50	0	0
Barn2 (Left)	0	0	0	0	0	1.58	0	0

Table 7.7: Miscellaneous Attacks

Chapter 8 Conclusion and Future work

8.1 Summary

In this thesis, a method of for determining the distortion caused by disparity based image rendering was implemented. Prior to presenting the proposed system, three theory-based chapters were introduced. The first few chapters provided the necessary background to understand disparity map estimation and image watermarking processes. An overview of the existing work done regarding 3D image watermarking was presented to demonstrate the various approaches used by researchers in this area. A few robustness enhancing techniques were introduced to address some of the limitations of existing watermarking schemes. With the aid of these techniques, a system meant to detect the effects of rendering was introduced and were combined to create a robust watermarking system. From the tests recorded in this thesis, it was clear that the rendering prediction scheme was capable of negating the effects of disparity map rendering because no difference between the left image and the right image were observed when rendered.

The bit capacity of the proposed system varied between 38% and 88%. This variance is attributed to two factors. First, as the number of objects in the scene increases, so does the complexity of the depth map. This increase in complexity results in a larger number of occluded pixels, reducing the available space in the decision matrix. The second factor that impacted this value is the cost function used. The cost function of choice determines the smoothness of the image. As the smoothness of the disparity map increases, the available space increases due to the constraints placed upon it, yet the fidelity of the watermarked image behaved as expected. When an embedding strength of zero was selected, the image block underwent the DCT process. This helps identify the noise that the DCT process introduces.

The system proved to be fairly robust to salt and pepper noise. It was not as effective against Gaussian noise, however. To help create realistic results, the majority of the attacks ensured that the PSNR between the watermark image and the attack image were greater than 35 dB. However, due to the high variability in the image , remaining above the 35 dB limit was not always possible. The most interesting observation is that the system was capable of withstanding standard filtering attacks. It

proved more robust against larger filter sizes, but it faced the greatest challenge in overcoming median attacks.

Due to some of the robustness enhancing techniques, the system was capable of overcoming histogram equalization and cropping. However, when JPEG compression was applied, a bit error rate of 10% was observed. Otherwise the system was capable of withstanding other compression ratios. This can be attributed to the quantization stage of the watermarking process. When compared to other disparity map-based watermarking schemes the system presented in this thesis was capable of:

- Embedding watermark in Left and Right Image,
- Reducing necessary information for the extraction process,
- Predicting distortion to limit watermark damage.

The proposed system was implemented and tested in MATLAB. From the results of the tests, the proposed system is capable of withstanding image-based rendering as well as a few watermark attacks. The tests performed used the Middlebury data set along with the Tsukuba data set. Due to the computationally intensive nature of disparity map imaging, the tests were limited to smaller images. However, the time performance of the disparity map estimation process can be significantly increased through the use of parallel processors, such as GPUs.

8.2 Future work

The work in this thesis focused on the watermarking process and omitted some of the more advanced techniques used in the rendering process, primarily hole-filling in the rendered image. Due to the constraints placed on the decision matrix, the hole-filling process would not have affected the watermark detection. Future work in this area could relax the constraints and attempt to create a mechanism that could survive such attacks.

Another worthy observation made during the course of this research, is that 40-50% of the pixels within the tests image correspond to the background. If the decision matrix is used to identify background pixels once the non-background pixels are removed, the cropped image can be scrambled and the resultant image can be watermarked. Once the image is unscrambled, the non-background pixels are reintroduced into the image. This would be the equivalent of white Gaussian noise attack.

By creating a watermarking scheme that can survive significant Gaussian attacks, the above system could be realizable and would provide an improved performance, resulting in a robust and blind watermarking scheme, that reduces the total amount of information required for the extraction process.

References

[1] Sin-Joo Lee and Sung-Hwan Jung, "A survey of watermarking techniques applied to multimedia", in *Industrial Electronics, 2001*. Proceedings. ISIE 2001. IEEE International Symposium on. IEEE, 2001, vol. 1, pp. 272–277.

[2] Jian Zhao, "Applying digital watermarking techniques to online multimedia commerce", in *Proc. Int. Conf. on Imaging Science, Systems and Applications* (CISSA'97). Citeseer, 1997, vol. 7.

[3] Vidyasagar M Potdar, Song Han, and Elizabeth Chang, "A survey of digital image watermarking techniques", in *Industrial Informatics*, 2005. INDIN'05. 2005 3rd IEEE International Conference on. IEEE, 2005, pp. 709–716.

[4] Rhys Hawkins, *Digital Stereo Video: display, compression and transmission*, Australian National University, 2002.

[5] Gerd Bruder, Frank Steinicke, and Wolfgang Sturzlinger, "Effects of visual conflicts on 3d selection task performance in stereoscopic display environments", in *3D User Interfaces (3DUI)*, 2013 IEEE Symposium on. IEEE, 2013, pp. 115–118.

[6] Victor S Grinberg, Gregg W Podnar, and Mel Siegel, "Geometry of binocular imaging", in *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*. International Society for Optics and Photonics, 1994, pp. 56–65.

[7] Victor S Grinberg, Gregg W Podnar, and Mel Siegel, "Geometry of binocular imaging", in *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology.* International Society for Optics and Photonics, 1994, pp. 56–65.

[8] Douglas Lanman, Matthew Hirsch, Yunhee Kim, and Ramesh Raskar, "Content-adaptive parallax barriers: optimizing dual-layer 3d displays using low-rank light field factorization", in *ACM Transactions on Graphics (TOG)*. ACM, 2010, vol. 29, p. 163.

[9] George Lawton, "3d displays without glasses: coming to a screen near you", *IEEE Computer Society*, 2011, vol. 44, pp. 17–19,.

[10] Philip Benzie, John Watson, Philip Surman, Ismo Rakkolainen, Klaus Hopf, Hakan Urey, Ventseslav Sainov, and Christoph von Kopylow, "A survey of 3dtv displays: techniques and technologies", *IEEE* 2007, vol. 17, pp. 1647–1658,.

[11] Aljoscha Smolic, Karsten Mueller, Nikolce Stefanoski, Joern Ostermann, Atanas Gotchev, Gözde B Akar, Georgios Triantafyllidis, and Alper Koz, "Coding algorithms for 3dtv – a survey", *IEEE* 2007, vol. 17, pp. 1606–1621,.

[12] Cheolkon Jung and LC Jiao, "Disparity-map-based rendering for mobile 3d tvs", *IEEE*.2011, vol. 57, pp. 1171–1175,

[13] Cheolkon Jung and Licheng Jiao, "Reliable depth-image-based rendering using parameter approximation in mobile devices", 2010, vol. 7, *The Institute of Electronics, Information and Communication Engineers*, pp. 666–671.

[14] Anders Olofsson, *Modern stereo correspondence algorithms: investigation and evaluation*, PhD thesis, Master Thesis, Department of electrical engineering, Linköpings Universitet, 2010.

[15] Peter Kauff, Nicole Atzpadin, Christoph Fehn, Marcus Müller, Oliver Schreer, Aljoscha Smolic, and Ralf Tanger, "Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability", Elsevier 2007, vol. 22, pp. 217–234,.

[16] Boguslaw Cyganek and J Paul Siebert, *An introduction to 3D computer vision techniques and algorithms*, John Wiley & Sons, 2011.

[17] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", Springer. International journal of computer vision 47, no. 1-3 2002, vol. 47, pp. 7–42,

[18] Kyung-Hoon Bae, Jung-Hwan Ko, and Jung-Suk Lee, "Stereo image reconstruction using regularized adaptive disparity estimation", 2007, vol. 16, , International Society for Optics and Photonics, pp. 013013–013013.

[19] Kyung-Hoon Bae, Jung-Hwan Ko, and Jung-Suk Lee, "Errata: Stereo image reconstruction using regularized adaptive disparity estimation", 2008, , International Society for Optics and Photonics, pp. 039801–039801.

[20] Ingemar J Cox, Matthew L Miller, Jeffrey A Bloom, and Chris Honsinger, Digital watermarking, vol. 53, Springer, 2002.

[21] Mohammad Abdullatif, Akram M Zeki, Jalel Chebil, and Teddy Surya Gunawan, "Properties of digital image watermarking", in *Signal Processing and its Applications (CSPA)*, 2013 IEEE 9th International Colloquium on., 2013, pp. 235–240.

[22] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon, "Secure spread spectrum watermarking for multimedia", IEEE 1997, vol. 6, pp. 1673–1687.

[23] Abhijit Mitra, "On pseudo-random and orthogonal binary spreading sequences", 2008, vol. 4, , Citeseer, pp. 137–144.

[24] Mei Yu, Aihong Wu, Ting Luo, Gangyi Jiang, Wujie Zhou, and Songyin Fu, "A new stereo image watermarking method for 3d media", 2012, Elsevier vol. 29, pp. 2399–2404.

[25] Chunhua Bai, Mei Yu, Gangyi Jiang, Zhongju Peng, and Feng Shao, "New hvs-based stereoscopic image watermarking algorithm for 3d media", in *Advances on Digital Television and Wireless Multimedia Communications*,. Springer, 2012, pp. 500–507.

[26] Patrizio Campisi, "Object-oriented stereo-image digital watermarking", 2008, vol. 17, *International Society for Optics and Photonics*, pp. 24–29,.

[27] Ji-Won Lee, Hee-Dong Kim, Hak-Yeol Choi, Sung-Hee Choi, and Heung-Kyu Lee, "Stereoscopic watermarking by horizontal noise mean shifting", in *IS&T/SPIE Electronic Imaging*. *International Society for Optics and Photonics*, 2012, pp. 7–17.

[28] Min-Jeong Lee, Ji-Won Lee, and Heung-Kyu Lee, "Perceptual watermarking for 3d stereoscopic video using depth information", in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*. IEEE, 2011, pp. 81–84.

[29] Yu-Hsun Lin and Ja-Ling Wu, "A digital blind watermarking for depth-image-based rendering 3d images", 2011, vol. 57, IEEE, pp. 602–611.

[30] Yu-Hsun Lin and Ja-Ling Wu, "Unseen visible watermarking for color plus depth map 3d images", in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on,* 2012, pp. 1801–1804.

[31] Fan Sheng-Li, Yu Mei, Jiang Gang-Yi, Shao Feng, and Peng Zong-Ju, "A digital watermarking algorithm based on region of interest for 3d image", in *Computational Intelligence and Security* (CIS), 2012 Eighth International Conference on. IEEE, 2012, pp 549–552.

[32] Gaurav Bhatnagar, Sanjeev Kumar, Balasubramanian Raman, and Nagarajan Sukavanam, "Stereo image coding via digital watermarking", 2009, vol . 18, *International Society for Optics and Photonics*, pp. 012–021.

[33] Afef Chammem, Mihai Mitrea, and Françoise Prêteux, "Dwt-based stereoscopic image watermarking", in *IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics*, 2011, pp. 26–36.

[34] Phen-Lan Lin, Po-Whei Huang, and An-Wei Peng, "A fragile watermarking scheme for image authentication with localization and recovery", in *Multimedia Software Engineering*, 2004. Proceedings. IEEE Sixth International Symposium on. IEEE, 2004, pp. 146–153.

[35] Dong Choon Hwang, Kyung Hoon Bae, and Eun-Soo Kim, "Stereo image watermarking scheme based on discrete wavelet transform and adaptive disparity estimation", in *Optical Science and Technology, SPIE's 48th Annual Meeting. International Society for Optics and Photonics*, 2004, pp. 196–205.

[36] Dong-Choon Hwang, Kyung-hoon Bae, Jung-Hwan Ko, and Eun-Soo Kim, "3d watermarking scheme in stereo vision system", in *Proceedings of SPIE, the International Society for Optical Engineering. Society of Photo-Optical Instrumentation Engineers*, 2005, pp. 28–38.

[37] Dong-Choon Hwang, Kyung-Hoon Bae, Maeng-Ho Lee, and Eun-Soo Kim, "Real-time stereo image watermarking using discrete cosine transform and adaptive disparity maps", in *ITCom* 2003. International Society for Optics and Photonics, 2003, pp. 233–242.

[38] Chunlin Song, Sud Sudirman, Madjid Merabti, and David Llewellyn-Jones, "Analysis of digital image watermark attacks", in *Consumer Communications and Networking Conference* (CCNC), 2010 7th IEEE. IEEE, 2010, pp. 1–5.

[39] Sadik AM Al-Taweel, Putra Sumari, Saleh AK Alomari, and Anas JA Husain, "Digital video watermarking in the discrete cosine transform domain", 2009, *Journal of Computer Science* 5, vol. 5, p. 536.

[40] Rolf Johannesson and Kamil Sh Zigangirov, *Fundamentals of convolutional coding*, Wiley-IEEE press, 1999.