DEPTH MAP COMPRESSION BASED ON PLATELET CODING AND QUADRATIC CURVE FITTING

by

Han Wang

A thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa

Ottawa, Ontario, Canada

September 2012

Abstract

Due to the fast development in 3D technology during recent decades, many approaches in 3D representation technologies have been proposed worldwide. In order to get an accurate information to render a 3D representation, more data need to be recorded compared to normal video sequence. In this case, how to find an efficient way to transmit the 3D representation data becomes an important part in the whole 3D representation technology.

Recent years, many coding schemes based on the principle of encoding the depth have been proposed. Compared to the traditional multiview coding schemes, those new proposed schemes can achieve higher compression efficiency. Due to the development of depth capturing technology, the accuracy and quality of the reconstructed depth image also get improved. In this thesis we propose an efficient depth data compression scheme for 3D images. Our proposed depth data compression scheme is platelet based coding using Lagrangian optimization, quadtree decomposition and quadratic curve fitting. We study and improve the original platelet based coding scheme and achieve a compression improvement of 1-2 dB compared to the original platelet based scheme.

The experimental results illustrate the improvement provided by our scheme. The quality of the reconstructed results of our proposed curve fitting based platelet coding scheme are better than that of the original scheme.

Contents

List of Tables

List of Figures

List of Acronyms

MVC	Multiview Video Coding
MVS	Multiview Video Sequence
HVS	Human Visual System
ATTEST	Advanced Three-dimension Television System Technologies
MPEG	Moving Picture Experts Group
JPEG	Joint Photographic Experts Group
PSNR	Peak Signal-to-Noise Ratio
bpp	bits per pixel
MSE	Mean Square Error
GOP	Group of Pictures
MMSE	Minimum average distortion optimization problem
MMAX	Minimax distortion optimization problem
MLEX	Lexicographically optimization problem
CUDA	Compute Unified Device Architecture

This thesis is dedicated to my family

Acknowledgement

This thesis is completed with the great help of my supervisor, my colleagues, my friends and my family.

First I would like to give my deepest gratitude to Professor Jiying Zhao, who not only guided in completing this thesis but also encouraged throughout my academic program. He pointed the right direction with his rich experience when I had problem with my project.

I would also like to thank our visiting professor, Dr. Fenghua Guo. Her previous research inspired me of completing this thesis. My final project would not completed so much without her help.

I want to thank all my lab colleagues, Wenyi Wang, Zhenyi Luo, Yu Zhang and Chengcheng Hao, and also all my friends in Ottawa. I spent the most wonderful and happy two years with you.

Finally I want give my most special thank to my family, from whom I feel the strongest and warmest support. All my success also belong to you, my dear father and mother.

Chapter 1

Introduction

1.1 3D Representation Technology

3D representation technology tries to show the viewer a new experience of viewing an image or a video. As one of the most important part of realizing this technology, efficient compression algorithm makes transmission possible. Recent years, lots of efforts have been made to achieve a high quality level of compression results. The size of the representation data will be different due to the differential used in different 3D equipments. Up to now the representation methods can be divided into two main directions. One is direct data presentation, such as Multiview video, video with associate data and the other is 3D mesh data [1].

The 3D mesh representation technologies are used to show the shape of the objects, no matter whether it is static or dynamic. This kind of technology is usually used in the computer technology. 3D mesh representation technology uses polygon meshes to represent the objects in the view. And one of the most important properties that indicates the calculation ability of a GPU in the computer system is the amount of polygon meshes it can generate every second. Three-dimension triangle meshes are used to form the targets' shapes on the screen. However,

although the 3D mesh representation technology can represent the shape of the target, it is very difficult to represent the distance of the target to the screen. What the technology can present to the viewer is still a 2D image or video.

The other representation method is the direct data presentation method. This kind of representation method tries to imitate the Human Visual System (HVS), represent two or more views at the same time and allow viewer to see a real "3D" image.

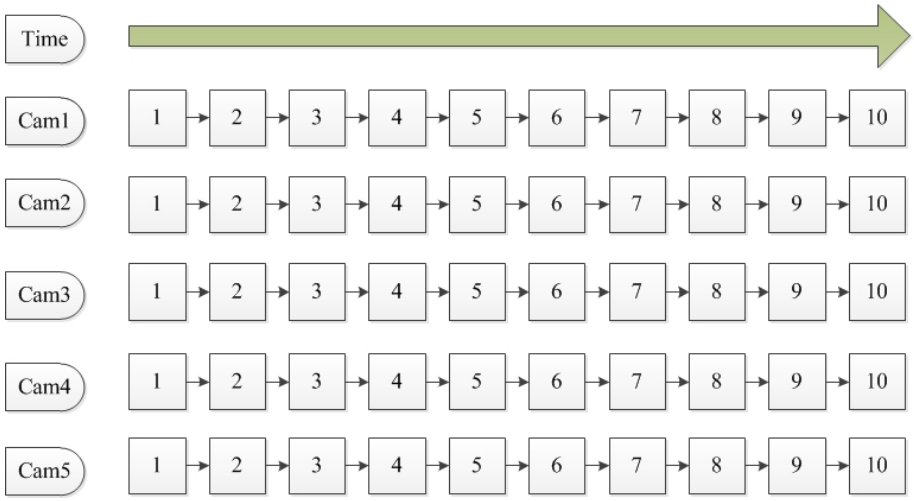


Figure 1.1: Multiview video sequence.

The Human Visual System uses two different views (eyes) to tell the location of the target and inform a 3D space to make the viewer think "this object is closer to me than that one". That is why many 3D videos are represented by different scenarios. It is captured originally by using several cameras to get the full information of the object and then transmitted to the receiving terminal. The receiving terminal receives those sequences and merges it to a stereoscopic image or video. This technology is commonly used in movie industry to film a 3D movie. The simplest case is the classical stereo video with only two cameras, which can be considered as an imitation of human eyes. But in order to capture as much information of the target as possible, more complicated system will contain 8, 16 or even more cameras.

This system captures a Multiview video sequence. The Multiview Video Sequence can be simply represented in Figure 1.1.

After generating those multiview video sequences, system will encode and transmit those video data to the receiving side. Decoder at the receiving side receives all this encoded bitstream and decodes them into different scenarios. Finally the rendering module generates a 3D represented video and presents this video to the viewer. The whole system is illustrated as in Figure 1.2

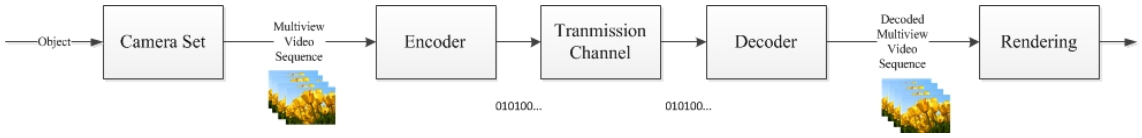


Figure 1.2: Multiview representation system.

1.2 Multiview Video Coding Technology

Using Multiview Video Sequence to transmit to the customer is a direct method in 3DTV broadcast system. The principle of encoding a set of Multiview Video Sequences is quite straight forward. It is by encoding all the video sequences of different view positions independently using the state-of-art codec algorithms such as H.264/AVC. However, the Multiview Video Sequences contain large amounts of inter-view dependence. The relative position of these cameras is fixed. This feature can be exploited for an encoding strategy combined of temporally and inter-view prediction. This encoding strategy is illustrated in Figure 1.3.

As illustrated in Figure 1.3, B-frames are not only predicted from temporally neighboring frames but also from adjacent view frames at the same time. It has been proved that using this method can get a significant compression rate improvement [2][3]. Thus compared to the direct "encode all the video sequences independently" coding scheme, the coding scheme

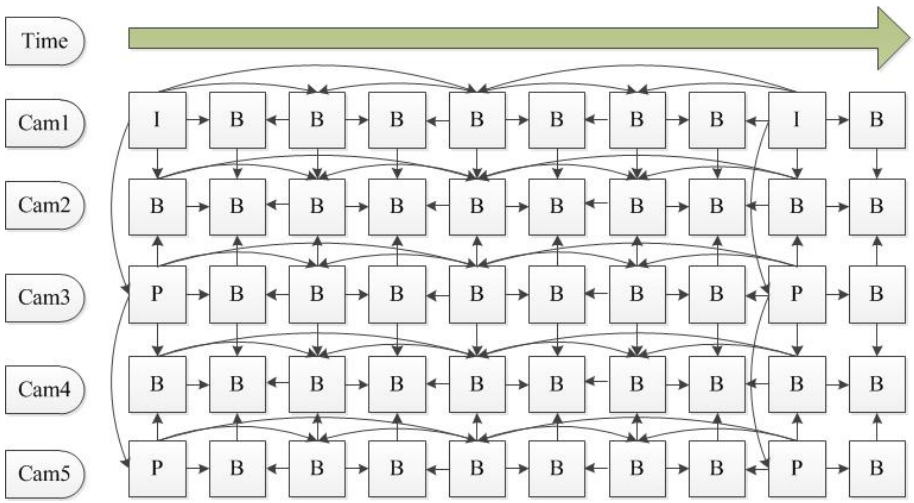


Figure 1.3: Multiview Video Coding (MVC) scheme.

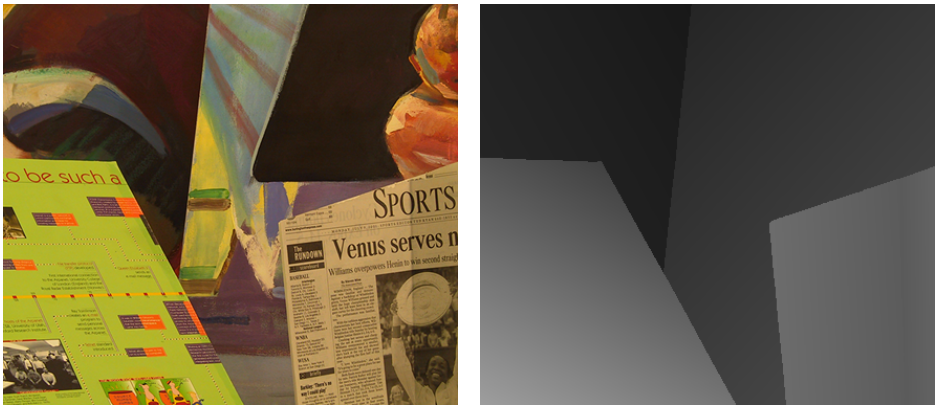
of MVC only needs to encode the I-frame, the motion vector and the disparity vector. This method significantly reduces the number of bits that are used to encode the Multiview Video Sequence.

Even though these state-of-art MVC algorithms can achieve a relatively higher coding efficiency compared to the direct "encode all the video sequences independently" way, the compression cost and complexity are still very high. Thus an alternative represent method is introduced to represent the 3D video or image. That is using associate data such as depth data to present the locations of the objects. In the transmitting procedure, system only needs to transmit the original video sequence and its corresponding depth data. The stereo video can be reconstructed from the video and the depth data at the receiving terminal. The depth data is stored as a grey level image which varies from 0 to 255. Because it only shows the distance information of the target object to the viewer's position, it has no relation to the texture of the object. This feature of the depth map makes high compression rate possible.

Several algorithms have been developed to efficiently compress the depth information. The European ATTEST project, which is stands for the Advanced Three-dimension Televi-

sion System Technologies, has proved that the depth data can be very efficiently compressed if embedding the depth data into the luminance channel of the video stream [4]. According to Smolic's conclusion [1], only 10% to 20% of the bit rate which is necessary needed to encode the color video is sufficient to encode the depth. An algorithm using depth-image-based rendering (DIBR) was proposed in 2004 by Fehn [5]. In his paper he also introduces the advantages over the classical approach of the stereoscopic video.

The earliest proposed algorithms encode the depth map data using the general video codec such as MPEG-4 or H.264/AVC. In 2005, Morvan *et al.* proposed a novel coding technique specific designed for depth images [6]. Depth image has a very important feature that the value of each pixel has no relation to the texture of the object at the same pixel. So even the original image could contain complex texture, its corresponding depth map can be very simple, as shown in Figure 1.4(a) and Figure 1.4(b).



(a) Color image:Venus (b) Depth image:Venus

Figure 1.4: Color image and its corresponding depth image.

We can notice that the original Venus image is made up with several newspapers and magazines, and the texture of this image is very complicated. But even though the texture on the newspaper and magazine is so complicated, the generalized depth image can be easily

divided into four main parts, two for the newspaper and magazine in the front, one for the background and one for the left part of the image. And for all four parts, it does not contain complex texture information such as the pictures or scripts on the newspaper. Thus the depth image is relatively simpler than the original image. Encoding the depth image can achieve higher efficiency than encoding the original image. By using this feature we can find an algorithm which is specifically designed to depth map and achieve a high compression ratio.

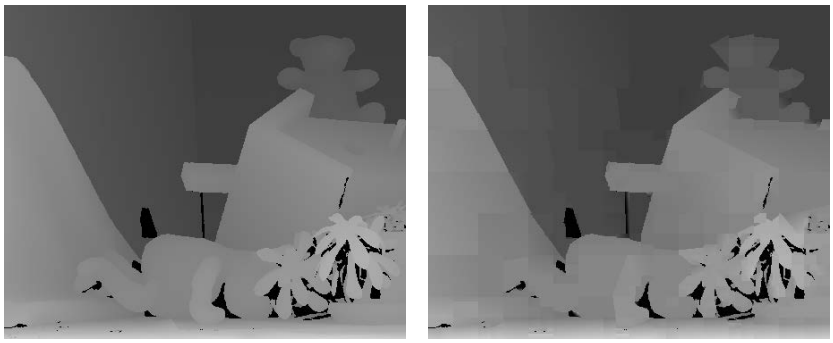
Another important feature of the depth data is that the depth image is linearly varied from the background to the foreground. This makes modeling the signal using piecewsie linear function possible. In 2005, Morvan divided the depth image with a quadtree decomposition principle and used functions selected from an independent linear function set to model each of the subdivided blocks. Morvan further developed his coding strategy in 2006 by introducing arithmetic coding technology into his previous work [7]. Generally the result of his coding algorithm outperforms a JPEG-2000 encoder by 1-2 dB.

Obviously Morvan and his work had shown a new direction in depth image coding aspect. Compared to other depth image coding scheme, Morvan's platelet based coding scheme uses the maximum statistical property of the depth map. Also his coding scheme can outperform the JPEG-2000 encoder at all level of the compression rate, while most of the other proposed depth map coding schemes cannot achieve such stability. Morvan's platelet based coding scheme can be considered as one of the most efficient coding schemes in depth image coding.

However there still exists some disadvantages in Morvan's algorithm.

- First of all, in order to reduce the algorithm complexity, Morvan uses straight line to subdivide a complex block. This compromises the preservation of edge information of the reconstructed depth image. We can see in Figure 1.5(b), the ears of the teddy bear have been reconstructed as an artificial triangle. Even though the PSNR value of this reconstructed image is very high, less accuracy of object's edge such as this could still

cause serious quality loss in the rendered image or video in the rendering procedure of this 3D image or video.



(a) Teddy:Original (b) Teddy:Decoded

Figure 1.5: Groundtruth depth image and its decoded image by Morvan's scheme [7].

- Secondly, when searching the separating lines of each block, Morvan uses an exhaustive searching method. Exhaustive searching means system will test every possible combinations of the two points on the edge of the block to form the separating lines, and selects the best combination that fits the actual division of the block. For an input block with size of 16×16, exhaustive searching equals to 120 iterations of the testing procedure. The number grows even larger when the size is bigger. This searching method enlarges both the complexity and operation time of the algorithm.

In order to improve the performance of Morvan's work, we choose to use his work as the basis and try to solve those disadvantages. After studying the scheme proposed by Morvan, we found that those two disadvantages are caused by using the straight line to separate a complex block. In 2011, Guo *et al.* published her research result showing that using quadratic curve instead of straight line in compressing the haptic data can reach a higher compression rate and accuracy [8]. Haptic data is more complex than the separating information in the depth image. Thus we were wondering if Morvan's original work could be improved by

using quadratic curve fitting instead of straight line in his original scheme. After conducting several experiments using the new scheme of the quadratic curve fitting technology and platelet based coding technology, we proved that Guo's research can also be effective in depth data compression.

1.3 Contributions of This Thesis

In this thesis, we first introduce the platelet based compression method proposed by Morvan in 2006, and then we make an implementation of his scheme according to our understanding. Morvan does not mention any detailed description of implementation in his original paper, so our work makes contributions by analyzing each techniques he proposed and explaining how to implement each step of his scheme. In the first part, we will give a clear explanation of how each step affects the implementation result, how to set and adjust the parameters and what we will do to deal with some specific situations.

Then we will also propose our compression method based on platelet coding and quadratic curve fitting technology. In our scheme, an automatic method based on platelet coding and quadtree decomposition principle is introduced. We use edge detection instead of exhaustive searching to get the edge information of each target block. This can significantly reduce the time consumption of the algorithm. And we apply quadratic curve instead of straight line to fit the extracted edge information. This can achieve a better preservation of edge information compared to Morvan's work. In addition, when compared to Morvan's scheme, our proposed scheme can also achieve a higher PSNR value at the same compression ratio.

The performances of our proposed scheme are impressive. From the experimental result we can see that our proposed compression scheme can compress the depth image at high compression ratio while remaining high quality. The final results of our proposed scheme

shows the higher compression efficiency, the better preservation of the silhouette of the depth information, and a faster operation time cost.

We believe further development of our proposed platelet based coding scheme could achieve higher compression efficiency and faster operation time. We hope our work could provide enough support to make the platelet based coding method to be used in high quality real-time 3D video coding.

1.4 Thesis Structure

The rest of this thesis is organized as follows. In Chapter 2 we will introduce the current development and recent technology of the 3D compression technology; In Chapter 3, an introduction to the platelet-based coding method will be presented; Chapter 4 describes our proposed platelet coding scheme using quadratic curve fitting technology; Chapter 5 illustrates our experimental results, analyses of our proposed scheme and gives comparison to the original scheme; Chapter 6 concludes the thesis and gives some suggestions for our future work.

Chapter 2

Literature Review

2.1 Multiview Video Coding Technology

3D video compression technology focuses on improving the coding efficiency and the bitrate control technique of the compression algorithm. Several algorithms of Multiview Video Coding have been introduced to improve the compression efficiency and lower its complexity. The advantage of Multiview Video Coding technology is that it is based on the standard video compression schemes such as H.264/AVC or MPEG-4. The multiview video sequence can be treated as a set of several video sequences of cameras shooting the same object at the same time but with the different position. So the statistical features of each video sequence could be studied in order to improve the coding efficiency.

During the year 2006 and 2007, several research groups developed inter-view/temporal prediction structures to exploit all the statistical dependencies within the multiview video sequences so that they can make improvements on the performance of the MVC coding scheme. Most of their approaches are based on H.264/AVC video coding standard.

In 2006, Kalva *et al.* proposed a 3DTV video system based on H.264 view coding [9][10].

Similar scheme can also be found in Bilen's work [11]. Socek *et al.* presented an alternative way to transform coding and arithmetic coding used to code motion compensation residuals in H.264/AVC to decrease the encoding and decoding complexity [12]. Schwarz and his group investigated the H.264/MPEG4-AVC coding with hierarchical B-pictures and proposed a motion-compensated temporal filtering based coding scheme. Their scheme could achieve the structure coding gains of 1 dB at the expense of increasing coding delay [13]. Also two Korean researcher Oh and Ho proposed a novel lattice-like pyramid GOP structure based MVC coding scheme in 2006, their scheme performance also has about 1 dB gain compared to the H.264 anchor coding scheme [14]. These algorithms are all based on the H.264/MPEG4-AVC syntax and proved to have a better performance. Because all of those coding schemes are based on research result of the statistical dependencies between temporal and inter-view pictures, the achievable gain of those schemes strongly depends on the content and some experiment properties such as camera distance, frame rate and complexity of the content.

In 2007, Merkle *et al.* proposed a similar compression scheme to achieve high compression efficiency [15]. His compression method is developed from the multiple reference picture technique in the H.264/AVC video coding standard. Merkle's proposed scheme exploits the statistical dependencies from both temporal and inter-view reference pictures for motion-compensated prediction. Average gains of 1.4 dB PSNR are reported by implementing Merkle's coding scheme. His scheme then was selected as the MVC standardization draft.

Other approaches focus on improving the motion estimation and disparity estimation algorithms to reduce the compression time. Disparity estimation illustrates the relation between different adjacent cameras. A fast motion estimation method is introduced by Li *et al.* in 2008 to reduce the estimation complexity and coding operation time [16]. Shen *et al.* studied the

complexity of the disparity vector estimation and built up a view-adaptive estimation scheme to save estimation time [17]. Its experimental results show although a small loss is occurred on the PSNR value of the reconstructed sequence (average 0.04 dB loss), this algorithm can significantly reduce the encoding time by 86% on average.

In recent years, research on the Multiview Video Coding technology focuses on the Scalable Multiview Video Coding technique. The SMVC is developed based on the view scalability that enables the multiview video coding bitstream to be displayed on a multitude of different terminals and transmitted over network with varying performance attributes. Early research in this aspect is made by Japanese researchers. Shimizu *et al.* proposed a view scalable multiview video coding using 3D warping with depth map in 2007 [18]. Their proposed scheme uses techniques from the field of image-based rendering to predict the target pictures. The experimental results showed that the proposed scheme can achieve about 40% bit reduction. Garbas and Kaup showed their interest in enhancing coding efficiency in spatial scalable multiview video coding with wavelets [19]. They proposed a scheme that first uses motion compensated temporal filtering and disparity compensated view filtering to de-correlate the multiview data, and then applies a spatial wavelet transform for the spatial decorrelation. They can enhance the quality of the reconstructed video by up to 3 dB. In 2010, Zhang *et al.* proposed a new joint multiview video coding scheme by combining the Scalable Video Coding standard with the MVC standard together [20]. The scheme can achieve about 10 to 20% saving bitrate compared with conventional MVC-based coding scheme.

Recent MVC technology has achieved a good performance in keeping a high compression efficiency. However, due to the development of the depth capturing technology, the captured depth data contains more accurate location information, and more efficient coding schemes based on encoding the depth information have been proposed.

2.2 Depth Map Coding Technique

As we have introduced in the first chapter, other approaches on efficient 3D video data coding without using MVC coding scheme focus on finding an efficient way to encode the 3D associate data such as the depth data.

The depth data contains information that directly shows the distance of the target pixel to the viewer's position. The value of the depth data is stored as a grey level image that varies from 0 to 255. This feature allows some specific coding schemes of grey level image to be applied to encode the depth data. Those schemes are designed to achieve high efficient coding performance. Compared to the performance of the MVC technique, depth map coding technique could be more efficient and adaptive.

During coding the depth map image, several important features of the depth map should not be forgotten. First the depth maps have large homogeneous regions within scene objects. The other important feature is that at object boundaries, the grey value of the depth could be sharply changed [21]. In the procedure of 3D image rendering, one key factor that affects the quality of the rendered image is the accuracy of its corresponding depth map. It is because the depth information shows the exact location information of this 3D image. This means in a depth signal, the high frequency signal is as important as the low frequency signal and should not be omitted in the compressing procedure.

Early proposed algorithms used to encode the depth map data are still based on standard codec such as JPEG-2000 for still images and MPEG-4 or H.264 for video. For a 3D video with depth data sequence, its relative depth data could be treated as an independent and simpler video sequence. Thus we can use existing video codec structure to encode the depth data sequence. How to find an efficient way to encode the whole depth data sequence then becomes equivalent to finding an efficient way to encode the intra-frame of the depth data

sequence. First attempts are made by using JPEG-2000 compression scheme.

Early in 2001, Krishnamurthy *et al.* proposed a compression and transmission scheme of depth map for image-based rendering [22]. However, the compression performance improvement of their coding scheme was limited compared to the standard JPEG-2000 encoder. The average improvement in terms of PSNR could reach 1.1 dB. They made two improvements in their proposed coding scheme. One is using region-of-interest coding to obtain more accurately depth information, the other is reshaping the dynamic range of the depth map to reduce the errors in the areas of smaller depth. They also made the contribution of integrating their research result into a standard JPEG-2000 decoder.

The European ATTEST project is initiated to design a backwards-compatible, flexible and modular broadcast 3D-TV system in 2002 [23]. The projects ends in 2004 with proposing a multi-use 3D video processing chain. Some accurate 3D rendering technology have also been proposed. For example in 2004, Fehn proposed a depth map encoding and rendering based 3DTV technique [5].

More approaches in this direction have been proposed recently. In 2011, Fan *et al.* proposed a new approach to 3D depth map motion estimation and compensation for 3D video compression. Their work achieves a good performance in compressing 3D video with the format of signal view plus depth data [24]. Last year, a novel rate control technique is presented by Liu *et al.* to accurately control the bitrate in order to satisfy the requirement of different 3D-video systems [25].

In 2005, Morvan *et al.* introduced a novel platelet based coding scheme specificly designed for the depth image and improved the coding efficiency compared to the traditional JPEG-2000 encoder [6]. In 2006 he went further to achieve an improvement of 1-2 dB compared to the JPEG-2000 encoder [7].

Morvan's work is based on a quadtree image segmentation technique and use the advan-

tage of the feature of the depth image that it contains large homogeneous regions within scene objects. This innovative attempt starts a new method to efficiently code the depth image. Recent years, those researches on image segmentation based compression of the depth image have reached great compression efficiency.

Zanuttigh and Cortelazzo proposed a similar image segmentation based compression scheme in 2009 [26]. They can reach an improvement of 3-5 dB for medium-high bitrates when compared to the standard JPEG-2000 scheme. And their proposed scheme not only offers the significant gain over JPEG-2000 but also produce high quality depth map without edge artifacts. However, their coding scheme still has limitations at low bitrates. In 2010, Graziosi *et al.* proposed an improved compression scheme using multiscale recurrent pattern image coding technique, which can significantly improve the PSNR for some specific depth image [27]. In one test image, their proposed scheme could reach a 10 dB improvement in PSNR compared to standard coding schemes. However this excellent performance could not be generally achieved. In some other experiments conducted by them, the performance of this scheme is not so excellent, the improvement is less than 1 dB in PSNR. In 2012, a faster compression scheme based on Morvan's work is presented by Sebal *et al.* [28]. Sebal rises the searching speed of the boundary in the depth map by using the edge detection technique. His proposed scheme shows significant gain in shortening the encoding delay of the whole coding process. However it also sacrifices the coding efficiency. Compared to Morvan's original scheme, Sebal's scheme could be about 2 dB less when it encodes the depth image at the same bitrate.

Other approaches to improving the quality of depth data have also been made. In 2009, Sarkis and Diepold proposed a new scheme based on compressed sensing data to compress the depth map. This scheme preserves the sharp edge of the depth map and ensures the smoothness elsewhere [29]. Their proposed scheme leads to lower error rate at high com-

pression ratio when compared to the traditional JPEG or JPEG-2000 standard image coding scheme. Last year, De Silva *et al.* studies the method to minimize the compression artifacts which are caused by using existing codecs. By using an adaptive bilateral filtering technique at the decoder to pre-process the compressed depth image, the scheme successfully minimizes the artifact caused by those codecs. The average quality improvement compared to the state-of-art technique could reach up to 1.7 dB [30].

After comparing those proposed depth image compression schemes, we found that the performance of the scheme proposed by Morvan is most stable and efficient under most of the conditions. Zanuttigh's scheme does not perform well in low bitrate condition and Graziosi's work cannot be effective on every image. Thus we select to improve Morvan's scheme to seek further improvement.

Chapter 3

Platelet-based Coding Method

This chapter describes the principles and implementation of Morvan's scheme. Morvan's research studies the unique feature of depth image compared to normal gray level images. His experimental results show that the platelet-based coding method is a very efficient way to encode the depth image. His implementation provides us a better understanding of the depth image compression technology.

Morvan's scheme mainly consists of three important principles: the Rate-Distortion Optimization, the platelet-based modeling and the quadtree decomposition. Generally Morvan used quadtree decomposition to decompose the input depth image into several sub-blocks. And for each sub-block Morvan assigned a best modeling function choosing from a preset modeling function set. The total number and size of the sub-blocks were decided by the Rate-Distortion Optimization. All these three technologies will be introduced in detail one by one in this chapter. In this thesis, we use the depth image called *Teddy* as Morvan used in most of his papers. The size of this depth image is 450×375, and we pad it to 512×512 for easier quadtree decomposition.

The Rate-Distortion Optimization method we use is based on the principle for indepen-

dent condition, which is first introduced by Ortega and Ramchandran in 1998 [31]. We consider our optimization problem as independent case because the selection of modeling function in each block has no relation with other blocks. And in the quadtree decomposition procedure, we also make some adjustments to reduce the number of iterations needed in the whole decomposition procedure.

3.1 Rate-Distortion Optimization

The system Rate-Distortion Optimization method for image and video compression was first introduced by Ortega and Ramchandran in 1998 [31]. In Ortege's paper, he illustrated varies kinds of Rate-Distortion Optimization methods for all kinds of possible data sets. He divided the optimal problems into two categories: the independent problems and the dependent problems.

In the independent case, the rate r_{ij} and the distortion d_{ij} can be measured independently for each coding unit. That is the rate and distortion result of coding one sub-block *i* has no relation to affect the result of coding another sub-block $i+1$. One example of the independent Rate-Distortion Optimization case is DCT image coding. When allocates bits to different blocks in the DCT image coding scheme, each block is individually quantized and entropy coded, the coding result of one block will never affect the coding result of other blocks. Obviously our platelet based coding scheme models each single block independently, the results of our modeling operation on block A will never affect the result of modeling block B. Thus we consider our platelet based coding scheme as the independent case. In this thesis we will focus on demonstrating the optimization method used for the independent case.

While in the dependency case, the prediction of each coding unit may depends on other coding units. The specific relation between each coding units can be different. For example,

if we assume that each coding unit i is predicted from its previous coding unit $i-1$, then when we build the predictor for this case, we need to use the past quantized data such as the prediction result of unit $i-1$. The dependent case is also widely considered. For example, when we consider the optimization method for some classic statistical condition such as the Markov chain, we need treat it as the dependent condition.

As illustrated in this thesis, the coding rate and distortion are measured independently for each block, so we will only discuss the Rate-Distortion Optimization problem under the independent case in this thesis. Further information for dependency optimization problems can be found in Ortega's paper [31].

Back to the independent case, the only fact that can change the optimization operation is the different constraint conditions. It is obvious that the constraint condition could vary depending on researcher's requirements. There always exists a storage constraint optimization problem when researching a new compression scheme. When optimizing a compression problem, the compression rate of a scheme is constrained by the maximum number of bits that can be used in this scheme. Then we can consider total number of bits available as the budget *R* of the optimization problem. So what we need to consider in this case becomes how to distribute those limited bits into different coding units while minimizing the overall distortion. It brings us a more specific optimization problem in optimizing compression problem - the budget constrained allocation problem.

3.1.1 Budget Constrained Allocation

Typically the budget constrained allocation problem can be re-stated in mathematical way as follows.

To find an optimal quantizer, or operating point, $x(i)$ for each coding unit i so that

$$\sum_{i=1}^{N} r_{ix(i)} \leq R, \tag{3.1}$$

and some metric for the distortion of each block $f(d_{1x(1)}, d_{2x(2)}, \ldots, d_{Nx(N)})$ is minimized [31]. For each coding unit i, its optimal quantizer could be and only be one. So if we consider x is the set of optimal quantizers, $x(i)$ is representing the operating quantizer which is chosen for unit i. Thus in Equation (3.1), $r_{ix(i)}$ is representing the bitrate and $d_{ix(i)}$ is representing the distortion for each coding unit i at this optimal quantizer $x(i)$. R is representing the maximum number of bits can be used (budget).

The metric for the distortion of each block can be different considering user's requirements. For example, if we are interested in minimizing the overall distortion of the target signal (noted that the signal could be a voice, a video or an image, etc., but in our thesis we mainly focus on solving the optimizing problem of an image signal), we can have a metric like this:

$$f(d_{1x(1)}, d_{2x(2)}, \ldots, d_{Nx(N)}) = \sum_{i=1}^{N} d_{ix(i)}. \tag{3.2}$$

Or if we are just interested in minimizing the maximum distortion of each sub-block, Equation (3.2) can be changed into:

$$f(d_{1x(1)}, d_{2x(2)}, \ldots, d_{Nx(N)}) = \max_{i=1:N}(d_{ix(i)}). \tag{3.3}$$

In both Equation (3.2) and Equation (3.3), $d_{ix(i)}$ is representing the distortion of each coding unit i. The optimal approach of using Equation (3.2) is called the minimum average distortion problem (MMSE). Some of the most widely used measurements such as PSNR can be considered as this kind of approach. The MMSE approach optimizes the overall perfor-

mance of the target scheme. The optimization results of this approach show the best quality under the restrict of certain number of bits. Other compression schemes also use Equation (3.3) to encode the image, such as Schuster's work in 1997 [32][33]. Those approaches are called the minimax approach (MMAX). The MMAX approach optimizes the scheme by comparing the best result each different method can reach. The optimization results of this kind of approach show the best quality the target scheme could reach. Those two optimization approaches are simple and most widely used in the optimization problems under independent case.

As an extension to the MMAX approach, lexicographically optimal (MLEX) approaches which compare two solutions by sorting their distortion or their quantization indices are proposed [34]. This kind of method first considers which quantizer can get the minimum distortion and then sort the quantization indices of all the coding units from largest to smallest (consider the smaller the quantizer is, the finer the distortion is). Then it compares the sorted lists of these quantizers. We say that the one presented by the smallest "NUMBER" is the best solution in the MLEX approach.

For example, for each of the 4 coding units, we have four quantizers 1, 2, 3 and 4 to encode them. Then we have the following two allocations (2, 1, 3, 4) and (3, 2, 3, 3). So after sorting those two allocations we obtain the rearranged lists (4, 3, 2, 1) and (3, 3, 3, 2). Obviously $3332 < 4321$, which gives us the conclusion that the second allocation is better than the first one in the MLEX approach. This kind of approach considers not only the encoding cost, but also the global distortion it may reach. In some circumstances it can successfully avoid some redundancy of the bit allocation by some global assignment methods. As we just mentioned in this MLEX approach, the encoding cost of a "3332" distortion blocks may be smaller than the cost of the "4321" distortion blocks, even though in the MMSE approach the "4321" shows a better distortion.

3.1.2 Lagrangian Optimization

In our platelet based scheme, the most important factor to be considered are the accuracy and quality of the overall reconstructed depth image. Thus the MMAX approach which only illustrates the best quality one method could reach is not fit for our requirement. In fact, a classical solution for the budget constrained allocation problem described in Equation (3.1) is introduced even earlier than the time when the MMAX or the MLEX constrained conditions are begun to be considered. It was first introduced by Everett in 1963 [35]. His solution was based on the discrete version of Lagrangian optimization method. After being proposed, this optimization method has been used by numerous number of authors. In this section we will describe how this Lagrangian optimization scheme works.

Everett's basic idea is as follows. He introduces a non-negative real number Lagrange multiplier $\lambda \geq 0$ to the Lagrangian cost function

$$J_{ij}(\lambda) = D_{ij} + \lambda \cdot R_{ij}, \tag{3.4}$$

where D_{ij} stands for the distortion of reconstructed block i when using the quantizer q_j and R_{ij} stands for the corresponding bitrate it use under this condition.

The rate-distortion location of each quantizer for a coding unit i is clearly shown in Figure 3.1. The dash lines shows a "plane wave" of slope λ from left to right. So the procedure of minimizing the Lagrangian cost function Equation (3.4) can be equivalent to finding the point in this R-D characteristic plane that is "first hit" by this wave. The final result (which quantizer should be chosen) is clearly shown in the figure.

There also exists an extreme condition when $\lambda = 0$. Under this condition the Lagrangian cost function in Equation (3.4) is changed to $J_{ij}(0) = D_{ij}$. This is equivalent to minimize the distortion only. As condition changes, the optimal result also changes. For example, if

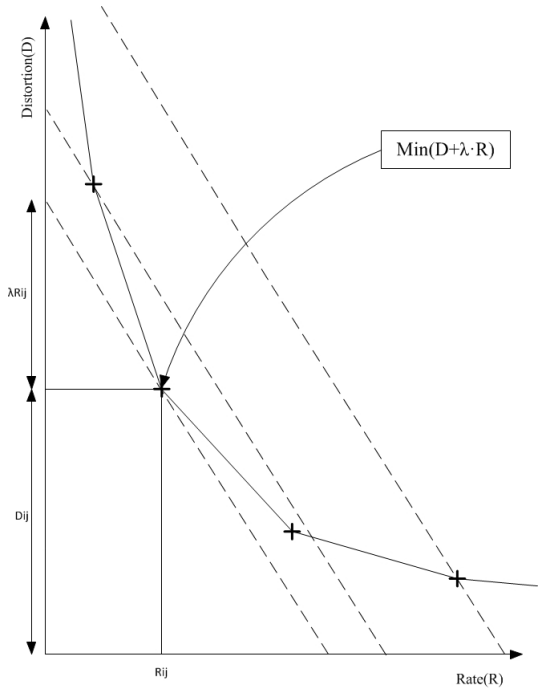


Figure 3.1: Rate-Distortion location of each quantizer for a coding unit i.

the four quantizers are still located at the same place in Figure 3.1, the best quantizer that minimize our new Lagrangian cost function should be chosen as the point closest to the Rate axis instead of the "$Min(D + \lambda \cdot R)$" point. And conversely, minimizing the Lagrangian cost function when λ becomes arbitrarily large is equivalent to minimizing the rate, i.e. selecting the point closest to the Distortion axis in Figure 3.1.

If the optimal quantizer $x(i)$ of each coding unit $i = 1, 2, \cdots, N$ in this Lagrangian optimization method minimizes the expression:

$$\sum_{i=1}^{N} d_{ix(i)} + \lambda \cdot r_{ix(i)}, \tag{3.5}$$

where λ represents the Lagrangian Multiplier, this quantizer set $x(i)$ is also the optimal solution to the budget constrained problem in Equation (3.1). The total budget R_T in this

condition is the sum of the bitrate of those coding units, it can be expressed by Equation (3.6)

$$R_T = \sum_{i=1}^{N} r_{ix(i)}. \tag{3.6}$$

This Lagrangian optimization method could constrain the global distortion of the whole coding object. The maximum global distortion will be no more than the sum of distortion of each coding units.

Using this classical solution based on Lagrangian optimization can successfully minimize the compression cost when the distortion is constrained.

In our proposed platelet coding scheme, this Lagrangian cost function (Equation (3.4)) is used when decompose the target image. During the quadtree decomposition step, the Lagrangian cost function is used as the necessary and sufficient condition in deciding whether the inputting block should be decomposed or not.

3.2 Platelet-based Fitting

Due to the feature that the grey value of the depth map has no relation to the texture of the objects and the feature of perspective, we can use a novel way instead of traditional MPEG-4 or H.264/AVC video coding method to encode the depth data.

All the functions used in this platelet based modeling scheme can be categorized into two classes: a class of piecewise-constant function and a class of piecewise-linear functions. First of all, the piecewise-constant function can be used to model some flat surfaces in the depth image. Those flat surfaces are used to indicate that this part of object is located with the same distance to viewer's position. The second class of functions can be used to approximate other regions which contain smooth change of the grey value. If the target block can be modeled by any function from those two classes, it can be represented by just a few coefficients. By

combining the modeling functions in this section and the Lagrangian optimization method introduced in the last section, we can efficiently decompose the depth image by a quadtree decomposition technique.

In this framework we use four modeling functions to predict the input block. Described as follows:

- Modeling Function $\widehat{f}_1$: Use a constant function to approximate the block
- Modeling Function $\widehat{f}_2$: Use a linear function to approximate the block
- Modeling Function $\widehat{f}_3$: Divide the block into two regions and approximate each region with a constant function (wedgelet function)
- Modeling Function $\widehat{f}_4$: Divide the block into two regions and approximate each region with a linear function (platelet function)

Note that in this thesis, our main improvement compared to Morvan's scheme is using a quadratic curve instead of a straight line to represent the separating line in modeling function $\widehat{f}_3$ and modeling function $\widehat{f}_4$. In order to illustrate how these modeling functions work in the implementation, we will use Morvan's scheme to demonstrate the framework in this section. Later in the next chapter, we will introduce our method's advantage and difference compared to Morvan's scheme.

3.2.1 Modeling Functions

The first modeling function $\widehat{f}_1$ can be used to approximate some smooth blocks. The modeling function can be simply represented as a constant function $\widehat{f}_1(x, y) = \alpha_0$.

The second modeling function $\widehat{f_2}$ is also used to approximate a smooth block. The difference between function $\widehat{f_2}$ and function $\widehat{f_1}$ is that function $\widehat{f_2}$ use a piecewise-linear function to approximate the block. Modeling function $\widehat{f_2}$ can be represented as

$$\widehat{f_2}(x, y) = \beta_0 + \beta_1 x + \beta_2 y, \quad (x, y) \in Block, \tag{3.7}$$

where $\{\beta_0, \beta_1, \beta_2\}$ represents the coefficient value of this modeling function. When it comes to a more complex condition, i.e. the target blocks include boundaries or edge, the third modeling function and the fourth modeling function can be used to model the input blocks. When using the third modeling function to model the input blocks, the block is separated into two regions A and B by a straight line. The sub-division line can be decided by listing all the possible lines in this block, then choosing the best one that can fit the boundary of the two regions.

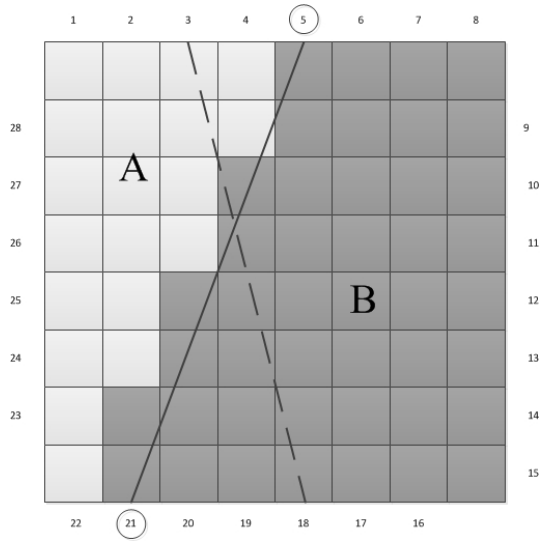


Figure 3.2: Illustration of how to represent the separating line.

For example, as illustrated in Figure 3.2, first of all, the system will label all the pixels located on the edge of the block one by one in a clockwise direction. Then it will choose two

pixels to form a separating line. Figure 3.2 shows how the separating lines would be when choose the combination of pixel 5 and 21 (straight line), or the combination of pixel 3 and 18 (dash line). The system only chooses one combination at one time. Obviously at this time the best combination fits the edge of the two regions is (5, 21. So when the system chooses the combination of (3, 18), it will move forward to test the next possible combination. This process will be stopped only after exhaustively searching all the possible combinations and finding out the best combination which fits the edge.

After determining the separating line, system will apply the first modeling method to each region of the block. The final modeling function can be represented as:

$$\widehat{f}_3(x,y) = \begin{cases} \widehat{f}_{3A}(x,y) = \gamma_{0A} & if(x,y) \in A \\ \widehat{f}_{3B}(x,y) = \gamma_{0B} & if(x,y) \in B \end{cases} . \tag{3.8}$$

where $\{\gamma_{0A}, \gamma_{0B}\}$ represents the coefficients used to model each sub-region A and B by this function. Equation (3.8) is called the wedgelet function, it can be used as the third modeling function in this scheme.

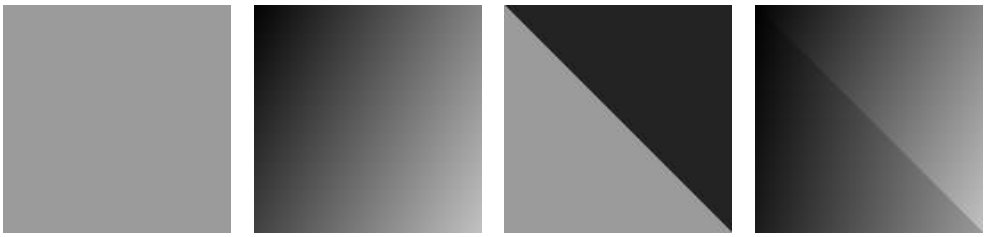
Similar to the wedgelet function, if applying the second modeling function $\widehat{f}_2$ instead of the first modeling function $\widehat{f}_1$ to each region of the block, the final modeling function can be represented as:

$$\widehat{f}_4(x,y) = \begin{cases} \widehat{f}_{4A}(x,y) = \delta_{0A} + \delta_{1A}x + \delta_{2A}y & if(x,y) \in A \\ \widehat{f}_{4B}(x,y) = \delta_{0B} + \delta_{1B}x + \delta_{2B}y & if(x,y) \in B \end{cases} \tag{3.9}$$

where $\{\delta_{0A}, \delta_{1A}, \delta_{2A}\}$ and $\{\delta_{0B}, \delta_{1B}, \delta_{2B}\}$ represent the coefficients used to model each sub-region A and B, respectively. That gives us the fourth modeling function, platelet function. Equation (3.9) shoes the platelet function. It can be used as the last modeling function in this

scheme to model the most complicated condition.

Finally, in order to give a full understanding of those four modeling functions, Figure 3.3 shows the examples of these functions.



(a) Function $\widehat{f}_1$ (b) Function $\widehat{f}_2$ (c) Function $\widehat{f}_3$ (d) Function $\widehat{f}_4$

Figure 3.3: Examples of those four modeling functions.

It is clear that for a certain input block, the reconstructed results of using those four different modeling functions are different. The coding scheme selects the modeling function based on only one rule. That is, the least bit cost is used to achieve the best quality.

After modeling the depth image block by block by using these modeling functions, the depth image can be represented only by the coefficients of those functions. The encoder only needs to encode the coefficients.

3.3 Quadtree Decomposition

In this section we will provide detailed information about how to implement the quadtree decomposition by using the optimal solution introduced in section 2.1. After implementing the quadtree decomposition, we can obtain an optimal result of several sub-blocks. The size of each sub-blocks are not the same, but each of them has been assigned to one of the modeling functions best fits the rate-distortion condition. Then those decomposed blocks will be sent to the encoder.

The most commonly used quadtree decomposition technique is called the *bottom-up* tree-pruning technique. This technique is proposed by Chou *et al.* in 1989 [36]. Detailed implementation method can be described like a true tree-pruning procedure. In our implementation we reverse the whole procedure to make the "pruning" technique becomes a "sprouting" technique. The principles of these two techniques are the same, so in this section we will introduce the principle of the pruning technique and at the end of this section we will introduce the difference between the pruning technique and the sprouting technique.

3.3.1 Pruning Technique

The pruning technique first fully segments the target image into a pixel level quadtree decomposition. All the segmented blocks have the same size of 1×1. Then the quadtree decomposition module will prune the nodes in this fully quadtree level by level to obtain an optimal quadtree decomposition of this image.

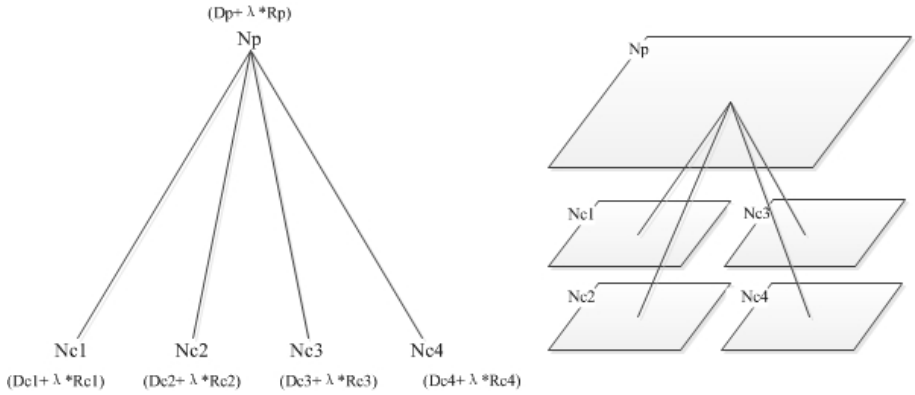


Figure 3.4: Illustration of the bottom-up pruning technique.

This technique is applied from the bottom of the tree to the top. As illustrated in Figure 3.4, the algorithm can be described by the following steps:

1. Choose four children nodes N_{c1}, N_{c2}, N_{c3} and N_{c4} generated by the same parent node

N_p.

2. For each child node a Lagrangian cost $(D_{N_{ck}} + \lambda R_{N_{ck}}), k \in 1, 2, 3, 4$, is calculated. Similarly the Lagrangian cost of the parent node, $D_{N_p} + \lambda R_{N_p}$, can also be calculated.

3. Compare the Lagrangian cost between the children nodes and the parent node. The four children node should be pruned if the sum of the four coding cost functions is higher than the cost function of the parent node. That is the children nodes we choose this time should be pruned when the condition in Equation (3.10) is met

$$\sum_{k=1}^{4}(D_{N_{ck}} + \lambda R_{N_{ck}}) > D_{N_p} + \lambda R_{N_p}. \tag{3.10}$$

The pruning procedure is designed as an iteration procedure starting from bottom to top. Note that during the iteration, if the selected node keeps its children nodes, then this node and its parent node will not be considered any more in the following iterations. Using this principle can significantly reduce the calculation time and cost. Finally one example of the pruned quadtree and its corresponding decomposition result of the target block are shown in Figure 3.5.

3.3.2 Sprouting Technique

The "sprouting" technique we used in this thesis uses the same principle illustrated in the previous section. The differences between implementing a sprouting technique and a pruning technique are as follows:

1. System selects one node as the parent node in one iteration, the corresponding children nodes of this selected node are then generated by the system temporarily. After generating those children nodes, the system will judge if the children nodes can be kept.

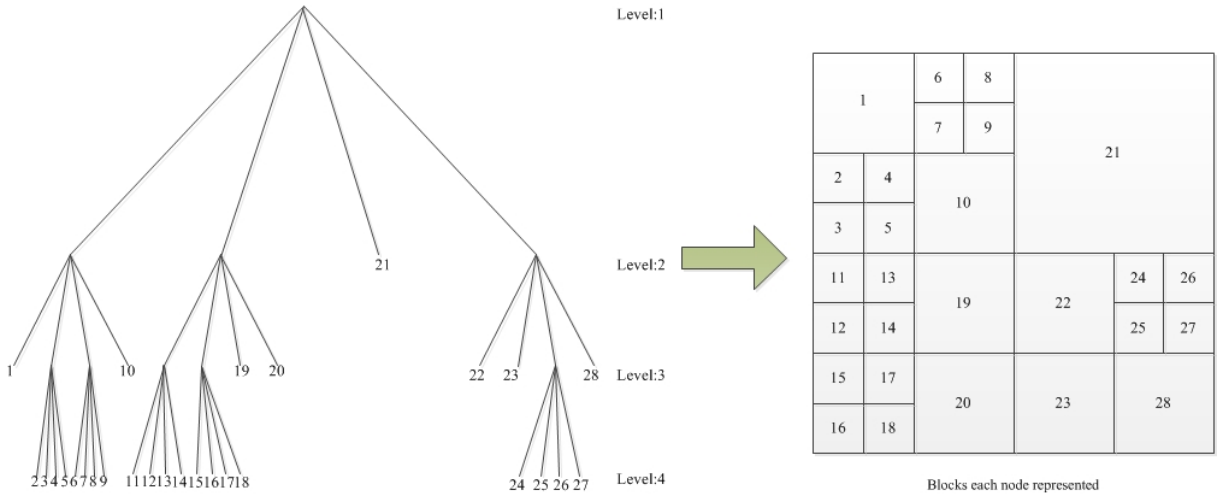


Figure 3.5: Pruned Quadtree and its corresponding decomposition result of the target block.

If and only if their Lagrangian cost fit the new judging equation described in Step 3 of this section, the children nodes can be kept;

2. Instead of applying a bottom-up iteration, the sprouting technique iterates from top level to the bottom level. For example, as illustrated in Figure 3.5, the iteration will start from Level 1 to Level 4;

3. After we calculate the Lagrangian cost value of all the parent node and children nodes, the children nodes can be kept if Equation 3.11 is met

$$\sum_{k=1}^{4}(D_{N_{ck}} + \lambda R_{N_{ck}}) \leq D_{N_p} + \lambda R_{N_p}; \tag{3.11}$$

4. If the selected node does not generate children nodes during an iteration, then in the next iterations this node or its corresponding children nodes will no longer be considered as a parent node.

Even though the sprouting technique and pruning technique are different, the optimal pruned tree we get from both techniques can be the same. The advantage of the sprouting technique is that it requires less memory to store temporary data during the implementation. Because the pruning technique is a bottom-up procedure, initial iterations require relatively more memory to store the information of nodes.

For example, when applying pruning technique to a full quadtree to get a pruned tree like Figure 3.5, the algorithm requires information of all 64 nodes of Level 4 as children nodes and all 16 nodes of Level 3 as parent nodes during the first iteration. So in order to implement the first iteration, pruning technique requires a minimum memory for storing $64 + 16 = 80$ nodes. During the next iteration we need 4 nodes of Level 3 as children nodes and 1 node of Level 2 as the parent node. The pruning technique does not require extra memory to store these nodes in this iteration. Thus the minimum memory requirement for pruning technique is 80 nodes. However, if we apply the sprouting technique, the first iteration of Level 1 needs information of only 1 node, the second iteration of Level 2 needs 4 nodes, the third iteration of Level 3 needs 12 nodes. The total memory requirement is just $1 + 4 + 12 = 17$ nodes.

During each iteration of implementing the quadtree decomposition by sprouting technique, each node of the current level has two status, one is the node needs to be further split and the other is it does not need to. Only the nodes do not need to be further split are our interested ones and need to be stored in the final result.

So after each iteration we only store the nodes labeled with "no-further split" tag to the final split result. Using the example pruned-tree in Figure 3.5, after first iteration only node No.21 is labeled with "no-further split" tag, so our algorithm will restore the information of block node 21 represented in the image (block size and upper-left point location).

An example of the fully optimized pruned quadtree of the *Teddy* image is shown in Figure 3.6. The Lagrangian Multiplier λ we choose in this condition is 100.

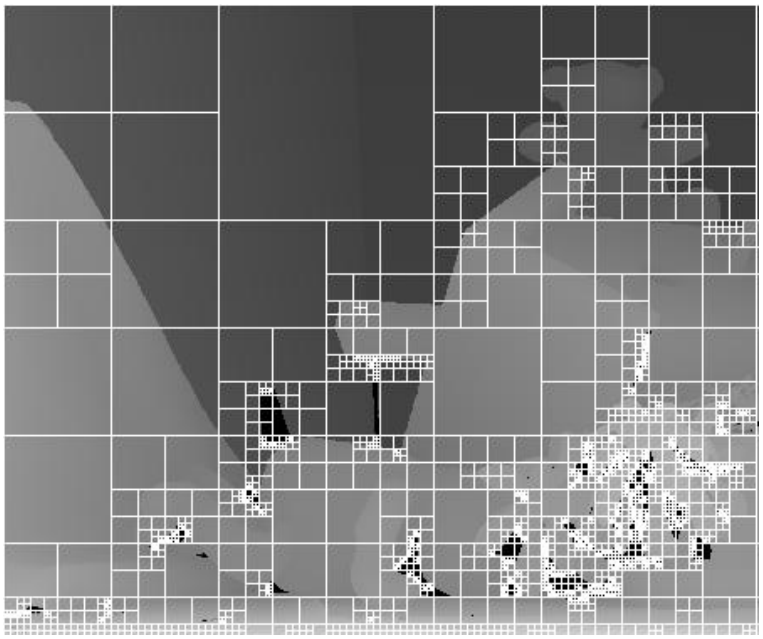


Figure 3.6: Split Teddy image at Lagrangian Multiplier value of 100.

For different Lagrangian Multiplier, the split results are different. The higher the value of the Lagrangian Multiplier is, the less the total number of the nodes of the optimized pruned tree has. In conclusion, by controlling the Lagrangian Multiplier, we can control the quality and size of the reconstructed depth image.

Chapter 4

Quadratic Curve Fitting based Platelet Coding Scheme

In this chapter we will describe in detail our proposed coding scheme based on quadratic curve fitting technology. According to Guo's research, using quadratic curve to fit the haptic data can improve data reduction rate and signal approximation precision [8]. This supports our goal to achieve a better preservation of the edge information of each block of the depth image. Also Guo's research supports one conclusion that when using quadratic curve to fit the target data, the number of curve segments will be less compared to the number of segments of using traditional straight line.

4.1 Algorithm Summary

Our proposed scheme uses these two advantages of quadratic curve fitting technology. This new proposed platelet coding scheme is based on the original scheme proposed by Morvan. The detailed encoding steps can be represented as the follows:

1. The two inputs to this scheme are, the depth image that needs to be encoded and a weight factor λ. This weight factor λ is used to indicate how important the compression rate R is in the Lagrangian cost equation. As illustrated in Section 2.1.2, if $\lambda = 0$, the Lagrangian cost based optimization will only minimize the distortion. As the value of λ becomes larger, the optimizer will consider the effect from the compression rate more.

2. Decompose the input depth image by using quadtree decomposition technique. The Lagrangian cost optimizer (λ) is used to determine the level of the decomposition. If we consider more effect from the compression rate, the level of quadtree decomposition will be less.

3. Approximate each of the decomposed sub-blocks by the four modeling functions $\widehat{f_1}$, $\widehat{f_2}$, $\widehat{f_3}$ and $\widehat{f_4}$. The output of this step is the approximated coefficients that needs to be encoded. While approximating the target blocks by wedgelet function ($\widehat{f_3}$) and platelet function ($\widehat{f_4}$), a faster edge detection technique is introduced.

4. Store the approximated coefficients generated from each block and encode them with high efficiency arithmetic coding scheme, then send the encoded bitstream through transmission channel.

The decoding procedure is almost the same as the encoding procedure. The bitstream that transmitted by the transmission channel contains the following information:

1. Size of each block. In order to decompose the original depth image more easily, we pad the original image into a square size image. Thus each block in the decomposition results are also with square size. We only need one number to present the size of each block;

2. Coefficients for each block. Those coefficients are key information to reconstruct the gray value of each block, this part of information includes both the coefficient used to fit the gray value of the block and the coefficients used to fit the separate curve in wedgelet function and platelet function. Detailed information about the coefficients can be found later in Table 4.2;

After receiving the bitstream at the decode terminal (customer terminal), decoder will decode the bitstream block by block and reconstruct the whole depth image.

In summary, the scheme structure can be represented in Figure 4.1.

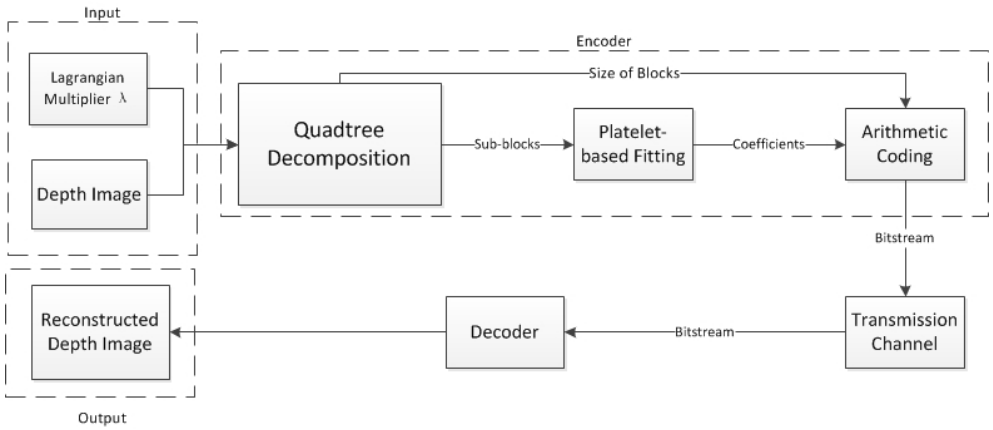


Figure 4.1: Summary of the codec structure.

In the rest part of this chapter, we will introduce our improvements in the quadratic curve fitting based platelet coding scheme compared to the original coding scheme.

4.2 Quadratic Curve Fitting

Quadratic curve fitting is used to achieve a high approximation precision. When compared to the result of using a straight line to fit the edge, the number of final nodes of quadtree is reduced.

Besides reducing the number of nodes, our quadratic curve fitting scheme also has other improvements compared to the original scheme. In our proposed quadratic curve fitting scheme, we use a faster edge detection technique and achieve a better preservation of the edge information.

4.2.1 Fast Edge Detection Technique

First of all, instead of exhaustive searching for the possible line to separate the block, we use a new method based on edge detection method to improve the searching speed. Similar scheme is proposed by Sebai *et al.* in 2012 [28].

In our scheme we choose to use the open operation to achieve a more accurate edge detection result. The open operation can be defined in Equation (4.1).

$$A \circ B = (A \ominus B) \oplus B, \tag{4.1}$$

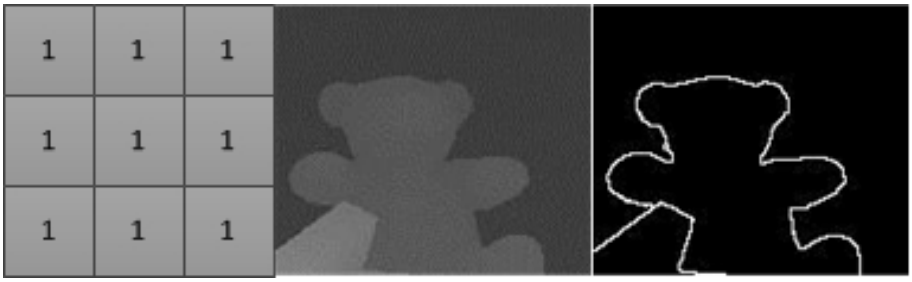
where $\ominus$ represents the eroding operation and $\oplus$ represents the dilation operation. Equation (4.1) shows the condition of using B to operate image A. Eroding operation and dilation operation are defined in Equation (4.2) and Equation (4.3), respectively

$$A \ominus B = \{a | B[a] \subseteq A, a \in A\}; \tag{4.2}$$

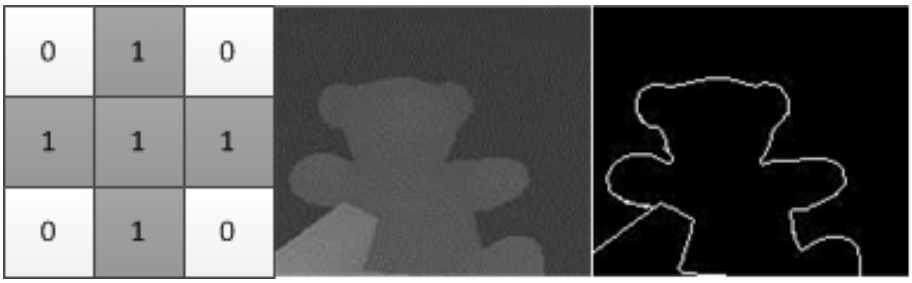
$$A \oplus B = \{a | B[a] \cap a \neq \emptyset\}. \tag{4.3}$$

Using different operator B results different edge detection result. Common operator shapes can be rectangle (Figure 4.2(a)) or diamond (Figure 4.2(b)).

In order to choose a relatively more precise edge detection method, we compared the re-



(a) Rectangle operator and its edge detection result

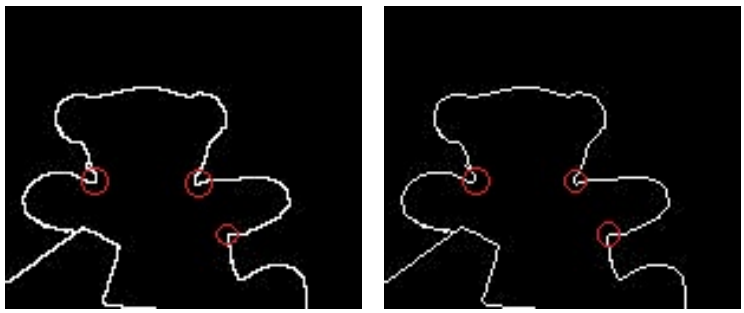


(b) Diamond operator and its edge detection result

Figure 4.2: Different operator shapes and its edge detection result.

sults obtained from different open operators. In Figure 4.3 we circled the differences between the result generated by the rectangle operator and the result generated by the diamond operator. After comparison, we find the result generated by the rectangle operator is less smooth than the result generated by the diamond operator. This may cause deviation when implementing our fitting technique. So we consider the diamond operator based edge detection method as the more precise detection method in our scheme.

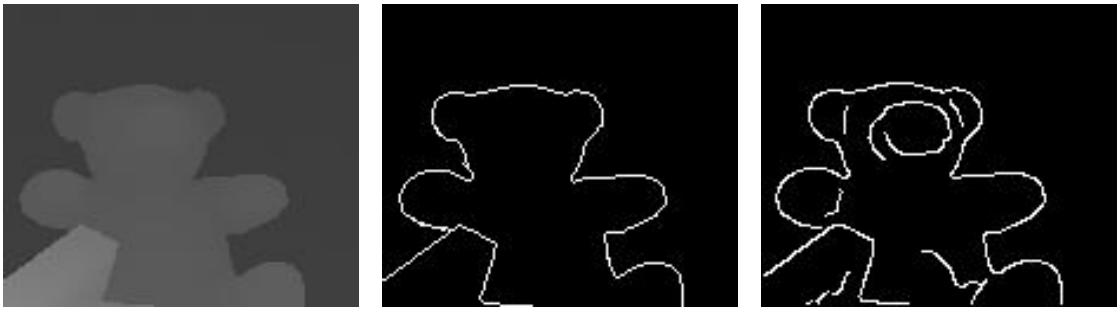
We have also compared our edge detection method with the most commonly used one, Canny method. As shown in Figure 4.4(c), the Canny edge detection method is too sensitive. The edge detection result generated by Canny method keeps redundant edge information such as those within the head of the Teddy bear. However what we are interested is only the silhouette of the Teddy bear. We need to adjust the parameters of the Canny method to reduce those redundant information. But for different blocks, the value of those parameters are different.



(a) Rectangle operator result (b) Diamond operator result

Figure 4.3: Comparison of results by diamond operator and rectangle operator.

This sensitivity restricts the performance of the Canny edge detection method in this scheme. So we select open operation method as our edge detection technique. Differences between the open operation based edge detection result and Canny method based result can be found in Figure 4.4.



(a) Original block (b) Open operation based result (c) Canny method based result

Figure 4.4: Comparison of results by open operation method and Canny method.

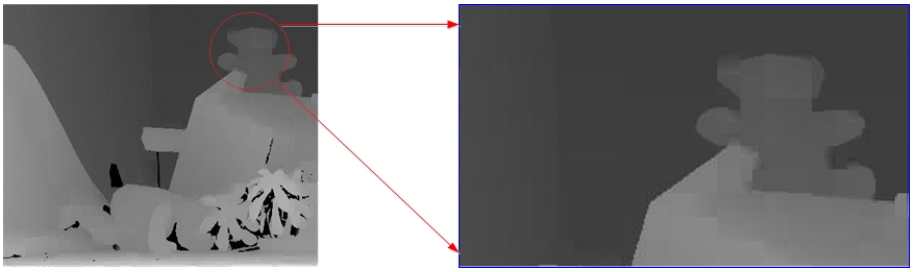
The edge detection result we obtained from using the Canny method contains redundancy information. Even though we can adjust the parameter in Canny method, the parameter we use to get an accurate detection result for one block could not be suitable with another block. Thus the adjusted parameter cannot be generally accepted to obtain accurate edge information in other blocks. After applying both two edge detection method we find using open operation

method can avoid this problem.

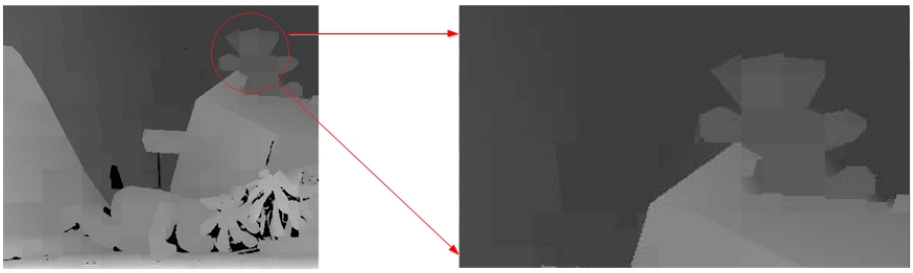
In conclusion, in order to obtain a relatively precise edge detection result, we choose the open operation method with diamond operator as our edge detection technique. The detected edge information is stored as coordination information in order to be further fitted by quadratic curve.

4.2.2 Preservation of Edge Information

One improvement of our proposed quadratic curve fitting scheme is the faster calculation speed. The other improvement is that the results of our proposed scheme have a better p-reservation of the edge information than the results of Morvan's original scheme.



(a) Quadratic curve fitting



(b) Exhaustive straight line searching

Figure 4.5: Comparison of results by quadratic curve fitting and exhaustive straight line searching.

The differences of those results are presented in Figure 4.5. Both of the figures indicate the decoded depth image after compressing the original depth image at 0.12 bpp (equivalent

to a compression ratio of about 66.7:1). Figure 4.5(a) has a PSNR of 36.35 dB, Figure 4.5(b) has a PSNR of 36.1 dB.

It can be clearly seen from Figure 4.5 that our quadratic curve fitting method gets a better preservation of the edge information. The silhouette of the bear reconstructed from our method is clearer and more precise than the exhaustive straight line searching method when both of the reconstructed images are at the similar compression ratio and quality.

Also as illustrated by Guo's research, when considering the same condition, which means the same dataset, the same perception threshold and the same linear prediction method, a quadratic curve fit provides a more accurate and smoother result than the linear prediction. The approximation precision can be significantly improved by using curve fitting technology. According to her experimental results, the MSE of curve fitting result could reach 5.00×10^{-5} while the MSE of linear fitting result under the same condition is 4.76×10^{-4}. The improvement of precision in the reconstructed data is almost ten times larger.

Both of our experimental results and Guo's experimental results show that using the quadratic curve fitting technology could actually improve the performance of achieving more precise edge information. Better edge information preservation in the reconstructed depth image means better performance when using this reconstructed depth image to render the 3D image. Thus in order to get a more precise reconstructed depth image, we choose the quadratic curve fitting technique to restore the edge information in each block.

By using the quadratic curve fitting technique to fit and restore the separating line (edge) of the original block, we can achieve more precise result with faster operation time. Also due to the fact that we reduce the total number of sub-divided blocks, the total bytes we used to store the coefficient of the quadratic curve could be less than the exhaustive straight line searching method. This improvement in compression efficiency is supported by our experimental result.

4.3 Estimation of Coefficients

When approximating the quadtree decomposed blocks in step 3 of the encoding procedure, we choose one of the four pre-defined modeling functions to get best approximating result. The encoding system will only transmit the approximated coefficients of each block to the arithmetic encoder.

4.3.1 Coefficient Estimation Method

The objective of this step is to obtain the best model and model coefficients to minimize the error between the original block and its corresponding approximation.

Coefficient estimation is straight forward when using a simpler modeling function ($\widehat{f_1}$ and $\widehat{f_2}$). The coefficient in $\widehat{f_1}$ simply corresponds to the mean value of pixels of the original block. This is the only way to minimize the error between original block and the reconstructed block by one coefficient. Thus the coefficient for first modeling function of an $n \times n$ block can be calculate by

$$\alpha_0 = \sum_{(x,y)\in Block} Block(x,y)/(n \times n). \tag{4.4}$$

Secondly, in modeling function $\widehat{f_2}$ is defined as in Equation (3.7), we need to estimate three coefficients β_0, β_1 and β_2. In order to approximate these three coefficients by the linear function defined in Equation (3.7), we use the least-square optimization. Because the value of pixel (x, y) in $\widehat{f_2}$ is represented as $\beta_0 + \beta_1 x + \beta_2 y$, we can use an equation set to calculate the coefficient sets β_0, β_1 and β_2. For an $n \times n$ block, this modeling coefficient set $\{\beta_0, \beta_1, \beta_2\}$ should minimize the error

$$Error_{\widehat{f}_2}(\beta_0, \beta_1, \beta_2) = \sum_{x=1}^{n}\sum_{y=1}^{n}(\beta_0 + \beta_1 x + \beta_2 y - Block(x, y))^2. \tag{4.5}$$

The ideal situation is that the value of each pixel of the original block can be determined by the same linear function $f(x, y) = \beta_0 + \beta_1 x + \beta_2 y$. In this situation Equation (4.5) reaches the minimum value 0, and the coefficients of this linear function are treated as the modeling result of this block. Thus approximating the coefficient of the original block can be equivalent to approximating the coefficient of this *Ideal Block*. This approximating equation set can be specified as:

$$\begin{pmatrix} n^2 & v & v \\ v & t & u \\ v & u & t \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} Block(x, y) \\ x \cdot Block(x, y) \\ y \cdot Block(x, y) \end{pmatrix}, \tag{4.6}$$

where $v = \frac{n^2(n+1)}{2}$, $t = \frac{n^2(n+1)(2n+1)}{6}$ and $u = \frac{n^2(n+1)^2}{4}$.

Thus if we have information of the size and pixel values of the original block, we can easily obtain the coefficient set $\{\beta_0, \beta_1, \beta_2\}$ by using Equation (4.6). This least square minimization based linear estimation can be equivalent to the optimization problem that using a linear plane to fit the original block.

And for the wedgelet modeling function and the platelet modeling function, we also need to determine and use extra coefficients to store the separating line (edge) in the original block. For each different separating line that separates the original block into two sub-region *A* and *B*, we can calculate the approximation error by summing the errors of both sub-regions. Our proposed quadratic curve fitting technique can find out the best coefficients that fit the separating line. This method also avoids a dilemma which may be occurred in the estimation procedure of this situation. That is, without knowing the subdivision boundary between the

sub-regions, it is impossible to determine the estimation coefficients for each sub-regions, but on the other hand, it is also not possible to identify the subdivision line as long as the sub-region coefficients are unknown. (Indicated by Morvan [7]).

In our proposed scheme, we first use quadratic curve fitting technique to determine the best separating line for the input original block, then we estimate coefficients for each sub-region to get the optimized approximation result.

After determining the best separating line for the original block, we can achieve two sub-regions. Then for each sub-region we can implement the same estimation method as modeling function 1 or modeling function 2. If both of the coefficients of the sub-regions are obtained by implementing modeling function 1, the final modeling function for the block is a wedgelet function ($\widehat{f_3}$). If both of the coefficients of the sub-regions are obtained by implementing modeling function 2, the final modeling function is a platelet function ($\widehat{f_4}$). In general, there are two coefficients $\{\gamma_{0A}, \gamma_{0B}\}$ for $\widehat{f_3}$ and there are six coefficients for $\widehat{f_4}$ which are divided into two subsets $\{\delta_{0A}, \delta_{1A}, \delta_{2A}\}$ and $\{\delta_{0B}, \delta_{1B}, \delta_{2B}\}$. According to what we have discussed above, using this estimation method can minimize the approximation errors in each sub-region. Because the original block is linearly combined by those sub-regions, minimizing the approximation errors of both sub-regions is equivalent to minimizing the error of the original block.

4.3.2 Summary

In summary, the coefficient estimation principle can be defined as in Table 4.1

Table 4.1: Coefficient estimation principle.

Modeling function	Coefficient estimation principle
Modeling Function $\widehat{f}_1$	Average value of the original block
Modeling Function $\widehat{f}_2$	Least square minimization based linear estimation
Modeling Function $\widehat{f}_3$	Average pixel values calculate separately for both region A and B
Modeling Function $\widehat{f}_4$	Least square minimization based linear estimation for both region A and B

Different modeling functions generate different number of coefficients. For the wedgelet function and platelet function, we need extra bits to encode the coefficient of the separating curve. Thus the total number of coefficients needs to be encoded is listed in Table 4.2.

Table 4.2: Number of coefficients needs to be encoded.

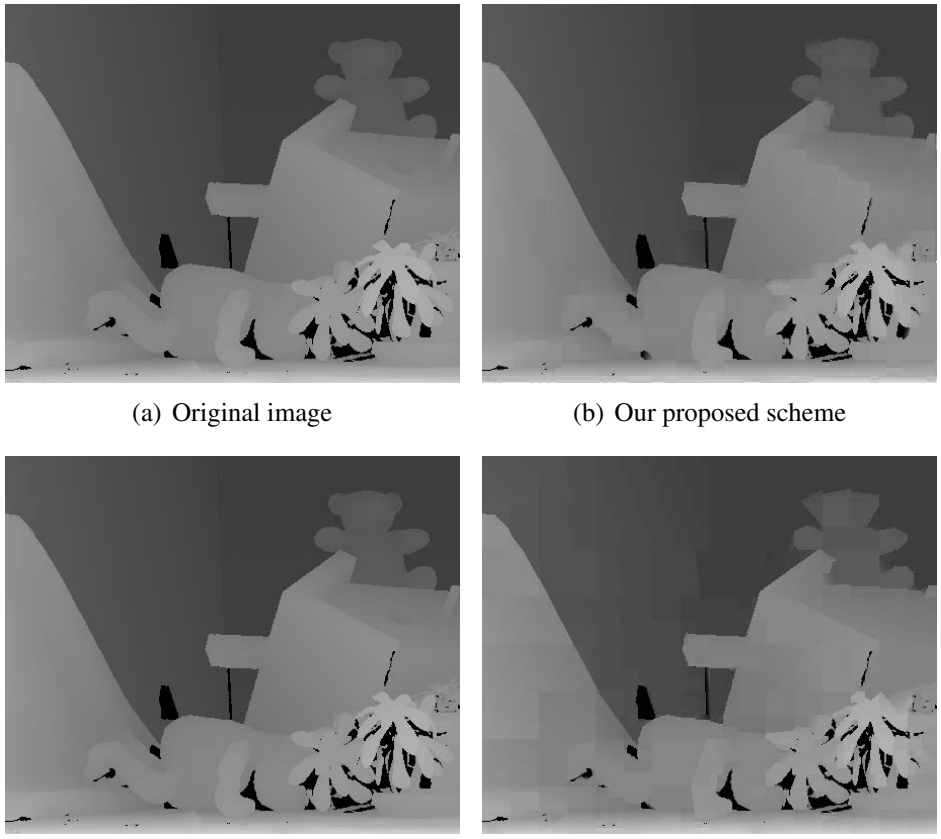
Modeling function	Coefficient	Coefficient for Separating line	Total Coefficients
Modeling Function $\widehat{f}_1$	1	0	1
Modeling Function $\widehat{f}_2$	3	0	3
Modeling Function $\widehat{f}_3$	2	3	5
Modeling Function $\widehat{f}_4$	6	3	9

The total number of coefficients need to be encoded is the same as the total number in the original platelet based compression scheme when using the modeling function 1 and modeling function 2, and the total number of coefficients is one more when using modeling function 3 and 4. However, because we have reduced the total number of sub-blocks, this increase in coefficients will not increase the total bits requirement for compression.

4.4 Improvements Compared to Original Scheme

Using our proposed coefficient estimation technique can obtain a better reconstructed depth image. Compared to Morvan's original scheme, our proposed scheme improves the quality

of the reconstructed depth image in many aspects. Those advantages we use make a more precise 3D image or video rendering technique possible. Also by improving the quality of the depth image we minimized the collimation error of the rendered 3D image.



(a) Original image (b) Our proposed scheme

(c) Original image (d) Morvan's original scheme

Figure 4.6: Comparison of reconstructed depth image by our proposed scheme and Morvan's original scheme.

The comparison of the reconstructed depth image between our proposed scheme and Morvan's original scheme is illustrated in Figure 4.6. Both of these reconstructed images are compared with the original *Teddy* image.

Both Figure 4.6(b) and Figure 4.6(d) are at the similar quality level (the PSNR value of Figure 4.6(b) is 36.35 dB, slightly bigger than the PSNR value of Figure 4.6(d), 36.1 dB) and with the same compression ratio. Both of the compression rate of Figure 4.6(b) and Figure

4.6(d) are at 0.12 bpp, which is equivalent to a compression ratio of about 66.7:1.

It can be clearly told that even both of the reconstructed images are at the similar quality level, our proposed scheme based reconstructed depth image reconstructed the original image more precisely. Also in Morvan's original scheme based reconstructed image, we can clearly notice some "blocks" on the background. This artefact error will affect the accuracy of the location information of the rendered 3D image. In our proposed scheme, the implementation of coefficient estimation technique eliminates these artefact errors.

Chapter 5

Experimental Results

In our proposed platelet based coding scheme, we implement both the original platelet coding scheme introduced by Morvan and our proposed platelet coding scheme with quadratic curve fitting technique. Our original depth images are ground truth depth images obtained from the database posted by Middlebury College [37]. Here we tested depth image "Teddy" and depth image "Cones". The "Teddy" image and "Cones" image are both grey level image at size of 450×375.

We implement the whole encoding and decoding process of both two platelet coding schemes and compare the performance of the decoded images at different compression rate. By adjusting the Lagrangian Multiplier λ, we can control the quality and compression ratio of decoded depth image. Different Lagrangian Multiplier determines different total number of nodes the quadtree decomposition technique finally generates. It is obviously that the more number of nodes the quadtree decomposition technique generates, the more precise and detailed information each node would be contained. In other word, it means the quality of the reconstructed depth image will be higher. Also the more number of nodes the quadtree decomposition technique generates, the more coefficients the approximation step will produce,

and more bits are needed to encode those coefficients.

It can be concluded that adjustment of the Lagrangian Multiplier would change the quality and compression ratio of the decoded depth image. And the quality of the depth image has negative relation with the compression ratio, i.e. the higher the quality is, the lower the compression ratio it will be. In all of the experimental result we present in this thesis, we use "bits-per-pixel (bpp)" as the parameter to represent the compression ratio. The relation of the compression ratio and the bpp can be present as $Compression\ Ratio = 8/bpp$.

5.1 Experiments on Image "Teddy"

The original "Teddy" image and the ground truth "Teddy" depth image (obtained by calibrated camera set, contains accurate depth information of the original image) is presented in Figure 5.1

(a) Teddy:Original (b) Teddy:Groundtruth

Figure 5.1: Original Teddy image and its corresponding ground truth depth image.

In the Lagrangian cost function $LagCost = Distortion + \lambda \cdot Rate$, $Distortion$ is represented by the MSE (Mean Square Error) value between the original depth image and the reconstructed depth image. And $Rate$ is represented by bpp, thus the compression rate of

each input node can be normalized considering the actual size of block. Table 5.1 listed the total number of nodes generated by both two platelet coding scheme when using the same Lagrangian Multiplier λ. The results in column "Original scheme" are obtained from our implementation of Morvan's original scheme.

Table 5.1: Comparison of total number of nodes generated by our proposed scheme and Morvan's original scheme (Teddy).

λ	Our proposed scheme	Original scheme
10	4621	5806
15	4309	5407
20	4135	5200
25	4048	5011
30	3948	4807
35	3883	4672
40	3847	4585
45	3799	4498
50	3751	4399
100	3280	3730
200	2956	3220
300	2806	2911
341	2770	2866
400	2611	2647
424	2566	2638
500	2419	2452
700	1852	1885
900	1555	1567
1100	1273	1210

We can tell from Figure 5.2 that when λ gets smaller, which means the quality of the image gets higher, the residual of the node number between these two platelet coding schemes become larger. Because the information contained within each separate nodes of these two schemes will tend to be the same when the reconstructed quality is low, the result of our proposed scheme and the original scheme will also tend to be the same under this circumstance.

When using large value of Lagrangian Multiplier, the number of nodes generated by our

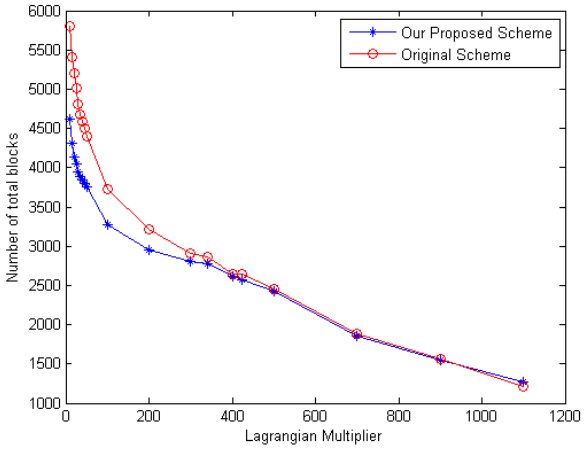


Figure 5.2: Number of nodes generated by our proposed scheme and Morvan's original scheme (Teddy).

proposed scheme could be slightly more than the number generated by Morvan's original scheme. For example, as shown in Table 5.1, by using the Lagrangian Multiplier value of 1100, the number of nodes generated by our proposed scheme is 1273, while the number of nodes generated by the original scheme is 1210. When the Lagrangian Multiplier is larger, the Lagrangian optimization would consider the compression rate more. In our proposed scheme, we added one more coefficient to fit the separating line. Thus when using Modeling function 3 or Modeling function 4 to fit the inputting block, the compression cost of this block by using our proposed scheme will be larger than the original scheme. Therefore under this condition, we tend to use simpler modeling functions in our proposed scheme to save the bitrate.

For example, assume that there exists an inputting block B which could be approximated either by using modeling function 1 ($\widehat{f}_1$) or by using modeling function 3 ($\widehat{f}_3$). When approximate it by using $\widehat{f}_1$, we need to decompose the block first, then fit each sub-blocks by $\widehat{f}_1$. In this example we use $LagCost_{our1}$ and $LagCost_{our3}$ to represent the Lagrangian cost of using $\widehat{f}_1$ and $\widehat{f}_3$ by our proposed scheme, respectively, and $LagCost_{ori1}$ and $LagCost_{ori3}$ to

represent the Lagrangian cost by the original scheme. Thus we have the following situations:

1. Clearly $LagCost_{our1} = LagCost_{ori1}$ because the distortion, compression rate and Lagrangian Multiplier of our proposed scheme and the original scheme are the same;

2. When using $\widehat{f_3}$ to fit the block, our proposed scheme needs 5 coefficients and the original scheme needs 4 coefficients;

3. When using $\widehat{f_1}$ to fit the block, both of our proposed scheme and the original scheme use total 4 coefficients;

By choosing the fitting method which generates smaller Lagrangian Cost, our proposed scheme will choose to use four $\widehat{f_1}$, and the original scheme will choose to use only one $\widehat{f_3}$. So the decomposition result of this block by our proposed scheme is 4 and by the original scheme is 1. That is the reason why when the Lagrangian Multiplier is very large, our proposed scheme generates more nodes.

By following the principle that choosing the method which has smaller Lagrangian cost, our proposed scheme and the original scheme will make different selection of the approximating method. Our proposed scheme will choose the approximating method with four $\widehat{f_1}$ because the Lagrangian cost is less, and the original scheme will choose the approximating method with only one $\widehat{f_3}$. So the decomposition result of this block by our proposed scheme is 4 and by the original scheme is 1. That is the reason why when the Lagrangian Multiplier is very large, our proposed scheme generates more nodes.

When our proposed scheme generates more nodes than the original scheme, the compression cost our proposed scheme used will be slightly higher than the original scheme. The quality of the reconstructed depth image by our proposed scheme is still higher than the result by the original scheme. When using the Lagrangian Multiplier value of 1100, the number of

nodes generated by our proposed scheme and the original scheme is 1273 and 1210, respectively. The quality of the reconstructed depth image by our proposed scheme is 29.3 dB at 0.0562 bpp, and the quality of the reconstructed depth image by the original scheme is 28.7 dB at 0.0511 bpp.

In conclusion, our proposed platelet based scheme can efficiently reduce the number of quadtree decomposition when using the same Lagrangian Multiplier.

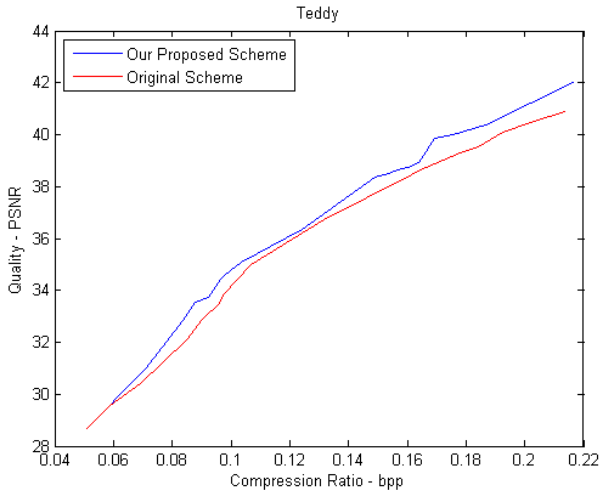
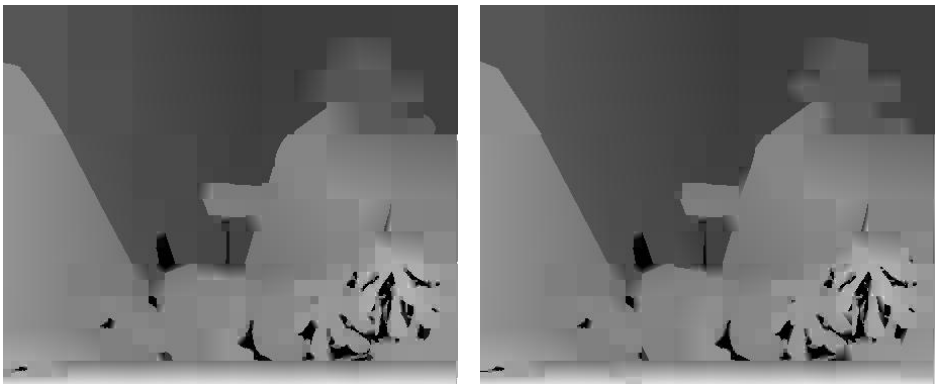


Figure 5.3: Comparison of Rate-Distortion performance of our proposed scheme and Morvan's original scheme (Teddy).

In our experiments we compare the performance of our proposed encoder and the original platelet encoder, results are illustrated in Figure 5.3. The experimental results in Figure 5.3 illustrate the PSNR values and bitrates of the reconstructed image by using Lagrangian Multiplier (λ) varies from 1100 to 10. When $\lambda = 10$, the reconstructed image quality of our proposed scheme is 40.3767 dB at 0.1865 bpp, and the quality of the original scheme is 40.9002 dB at 0.2136 bpp. Our proposed scheme can also reach the bitrate of 0.21 bpp when $\lambda = 5$, the quality of the reconstructed image is 42.0096 dB at 0.2164 bpp. And when $\lambda = 1100$, the reconstructed image quality by using our proposed scheme is 29.2832 dB at 0.0562 bpp and 28.6572 dB at 0.0511 bpp by using the original scheme.

We can concluded from Figure 5.3 that our proposed encoder consistently outperforms the original platelet based encoder. The improvements can reach as high as nearly 2 dB or 0.05 bpp for the "Teddy" in condition of higher compression rate (more than 0.17 bpp).

When the bitrate becomes smaller, the performance of our proposed encoder and the original encoder tends to be similar. If the bitrate is extremely low, the performance of our proposed encoder could be slightly worse than the original encoder. However when the bitrate is extremely low, the reconstructed depth image becomes heavily distorted (Figure 5.4). Such a heavily distorted depth image does not have much value.



(a) Our proposed encoder (28.4dB, 0.049bpp)

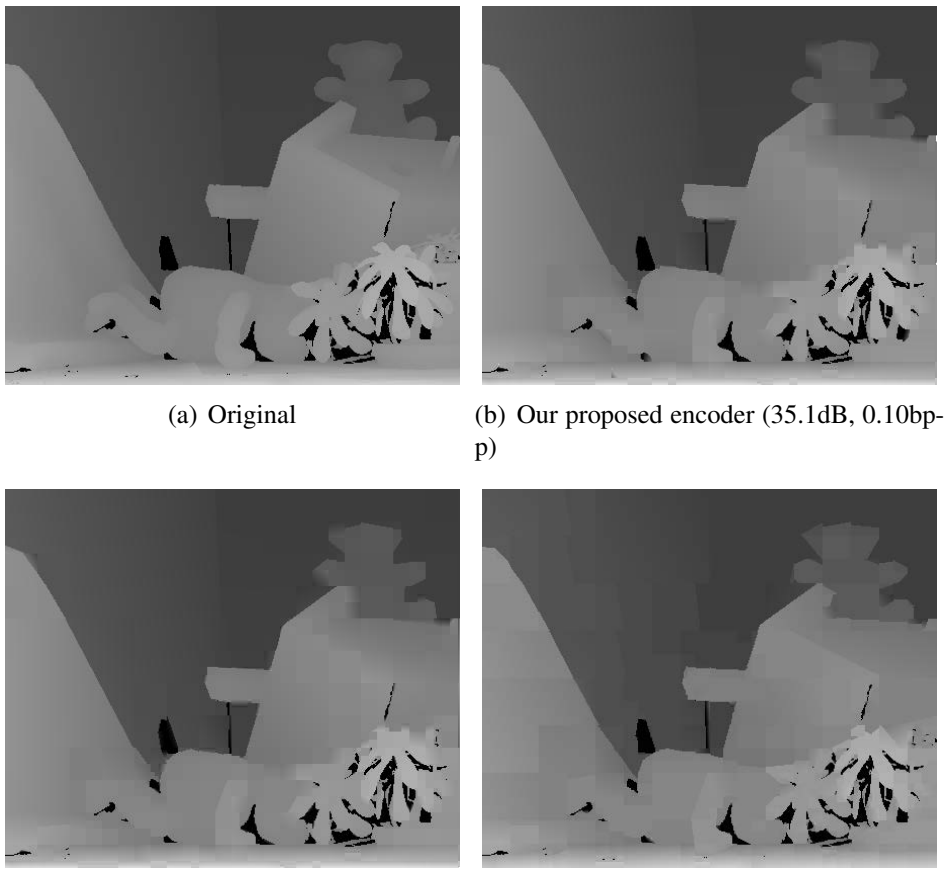
(b) Original encoder (28.6dB, 0.050bpp)

Figure 5.4: Reconstructed image at low bitrate by our proposed scheme and Morvan's original scheme (Teddy).

Finally we compare four depth images together to illustrate the advantage of our proposed encoder. Those four depth images are, the ground truth depth image, the depth image reconstructed by our quadratic curve fitting based platelet coding scheme, the depth image reconstructed by the original exhaustive searching based platelet coding scheme and the depth image reconstructed by Morvan in his paper. All the reconstructed depth images are at compression rate of around 0.11 bpp and PSNR of around 36 dB.

By comparing Figure 5.5(b) and Figure 5.5(c) we can conclude that our quadratic curve

fitting based platelet coding scheme keeps a better silhouette of the bear (The left hand of the bear in Figure 5.5(c) looks like to be "cut" by the encoder). By comparing Figure 5.5(b) and Figure 5.5(d) we can tell that our proposed encoder not only keeps a better silhouette of the image, but eliminates the artefact error occurring on the background of Figure 5.5(d).



(a) Original

(b) Our proposed encoder (35.1dB, 0.10bpp)

(c) Original encoder (35.0dB, 0.11bpp)

(d) Original paper (36.1dB, 0.12bpp)

Figure 5.5: Comparison of encoder performance of our proposed scheme and Morvan's original scheme (Teddy).

In the end of this chapter we illustrate all the results of our experiments on the depth image "Teddy" to show how the scheme works from the most quality condition to the most noisy condition (low quality) (Figure 5.13).

5.2 Experiments on Image "Cones"

Compared to image "Teddy", image "Cones" contains more geometry shapes and more complicated. The original "Cones" image and its corresponding depth image are presented in Figure 5.6

(a) Cones:Original (b) Cones:Groundtruth

Figure 5.6: Original Cones image and its corresponding ground truth depth image.

We also record the total number of nodes generated by the quadtree decomposition technique under different Lagrangian Multiplier. Detailed results are illustrated in Table 5.2 and Figure 5.7. The results in column "Original scheme" are obtained from our implementation of Morvan's original scheme.

Figure 5.7 shows the similar result as in Figure 5.2, thus we can conclude that our proposed scheme is also effective on the Cones image.

In order to prove that the advantages of our proposed scheme is not specific for the Teddy image, we also conduct several experiments on the "Cones" image by using both our proposed platelet based coding scheme and the original platelet coding scheme. The comparison of the PSNR and compression rate result of both scheme are shown in Figure 5.8.

Figure 5.8 also illustrates the PSNR values and bitrates of the reconstructed image when using Lagrangian Multiplier varies from 10 to 1100. When $\lambda = 10$, the reconstructed image quality is 40.9922 dB at 0.2097 bpp by our proposed scheme and 39.7669 dB at 0.2497 bpp

Table 5.2: Comparison of total number of nodes generated by our proposed scheme and Morvan's original scheme (Cones).

λ	Our proposed scheme	Original scheme
10	4708	6547
15	4522	6175
20	4312	5947
25	4141	5809
30	4057	5533
35	3973	5395
40	3913	5227
45	3847	5140
50	3811	5026
100	3280	4174
200	2602	3067
300	2089	2347
341	1996	2149
400	1882	1984
424	1708	1933
500	1657	1684
700	1330	1471
900	1162	1195
1100	970	1018

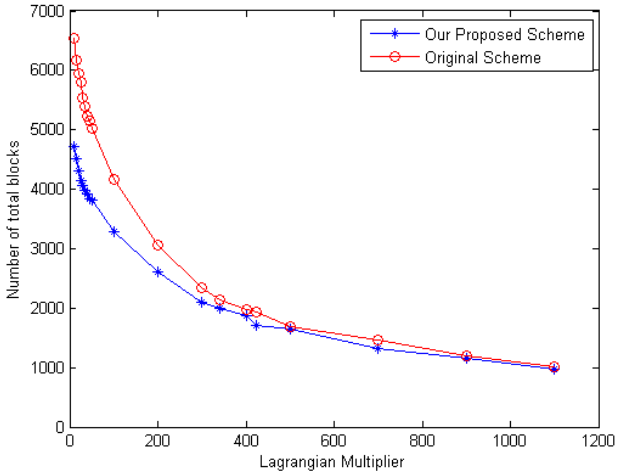


Figure 5.7: Number of nodes generated by our proposed scheme and Morvan's original scheme (Cones).

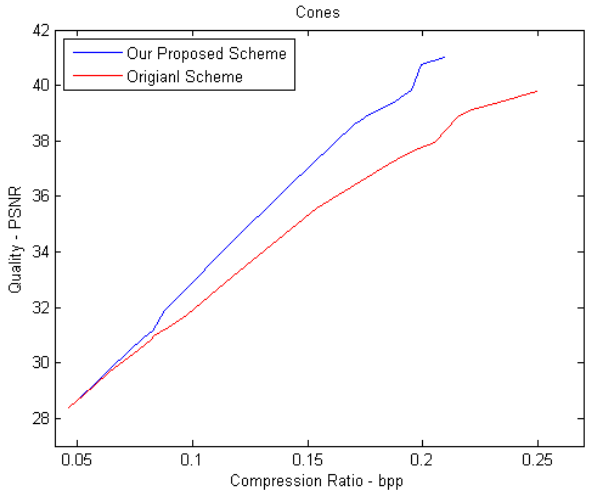


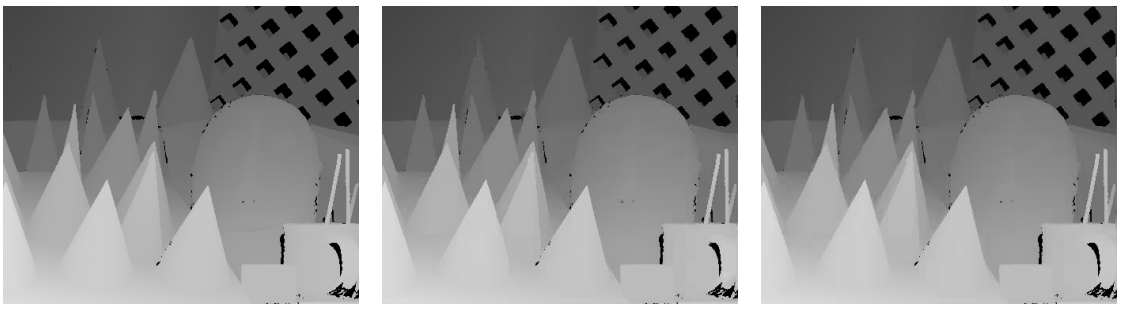
Figure 5.8: Comparison of Rate-Distortion performance of our proposed scheme and Morvan's original scheme (Cones).

by the original scheme. When $\lambda = 1100$, the reconstructed image quality is 28.7009 dB at 0.0501 bpp by our proposed scheme and 28.3677 dB at 0.0463 bpp by the original scheme. Similar to Figure 5.3, the performance of our proposed scheme will be slightly worse than the original scheme when the bitrate is extremely low.

Figure 5.8 shows that the improvement of the performance of our proposed scheme can reach to about 3 dB or 0.08 bpp for the "Cones" image in condition of more than 0.15 bpp compression rate. The improvement we get from the Cones image is even higher than the improvement we get from the Teddy image.

Also we choose the reconstructed images of these two schemes by using the same Lagrangian Multiplier $\lambda = 25$ to compare (Figure 5.9). We can hardly tell the differences between those two images from human visual. But both the PSNR value and compression rate show our proposed encoder achieve a higher compression efficiency. Our proposed encoder uses less compression rate ($0.186\ bpp < 0.215\ bpp$) to reach a better image quality (PSNR value $39.3\ dB > 38.8\ dB$)

In the end of this chapter we also illustrate all the results of our experiments on depth



(a) Original (b) Our proposed encoder (39.3dB, 0.186bpp) (c) Original encoder (38.8dB, 0.215bpp)

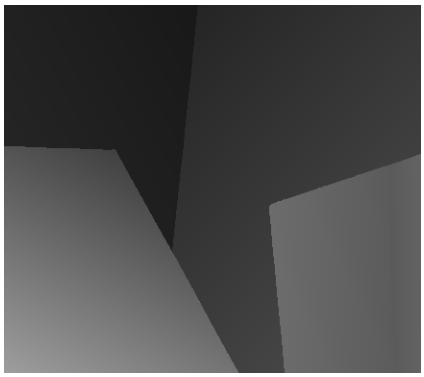
Figure 5.9: Comparison of encoder performance of our proposed scheme and Morvan's original scheme (Cones).

image "Cones" to show how our scheme works from the most quality condition to the most noisy condition (Figure 5.14)

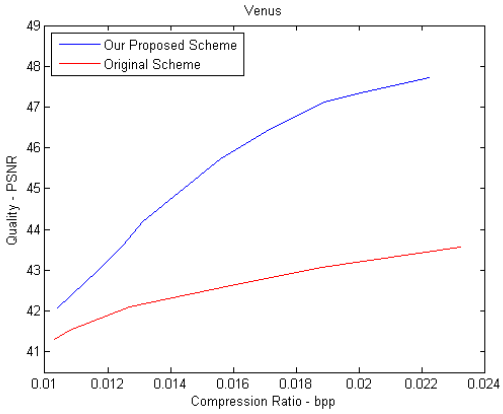
5.3 Other Experimental Results and Summary

Besides the "Teddy" image and the "Cones" image, we also tested other depth image such as the simple case "Venus" image (Figure 5.10(a)). The Venus image can be very efficiently compressed by platelet based encoding schemes. We also implement both of our proposed coding scheme and the original coding scheme to the image and compare their rate-distortion performance.

Figure 5.10(b) illustrates the PSNR values and bitrates of the reconstructed image by using several Lagrangian Multipliers varies from 10 to 400. When $\lambda = 10$, our proposed scheme reconstructs the depth image at quality of 47.7320 dB and bitrate of 0.0222 bpp, the original scheme reconstructs the image at quality of 43.5693 dB and bitrate of 0.0232 bpp. When $\lambda = 400$, our proposed scheme reach 43.0057 dB at 0.0117 bpp and the original scheme is 41.0622 dB at 0.0101 bpp. Our proposed scheme can also reach 0.01 bpp when



(a) Venus ground truth depth image

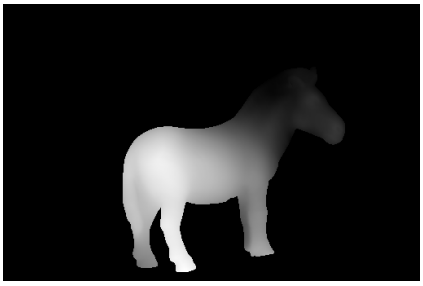


(b) Comparison of Rate-Distortion performance of the two schemes (Venus)

Figure 5.10: Depth image Venus and the comparison of Rate-Distortion performance of our proposed scheme with Morvan's original scheme (Venus).

$\lambda = 500$, the reconstructed image quality is 42.0676 dB at 0.0104 bpp. It can be observed from Figure 5.10(b) that our proposed scheme performs very well. For this kind of simple depth images, our improvement could be 5 dB higher even in a relative lower compression rate condition.

(a) Tsubuka ground truth depth image



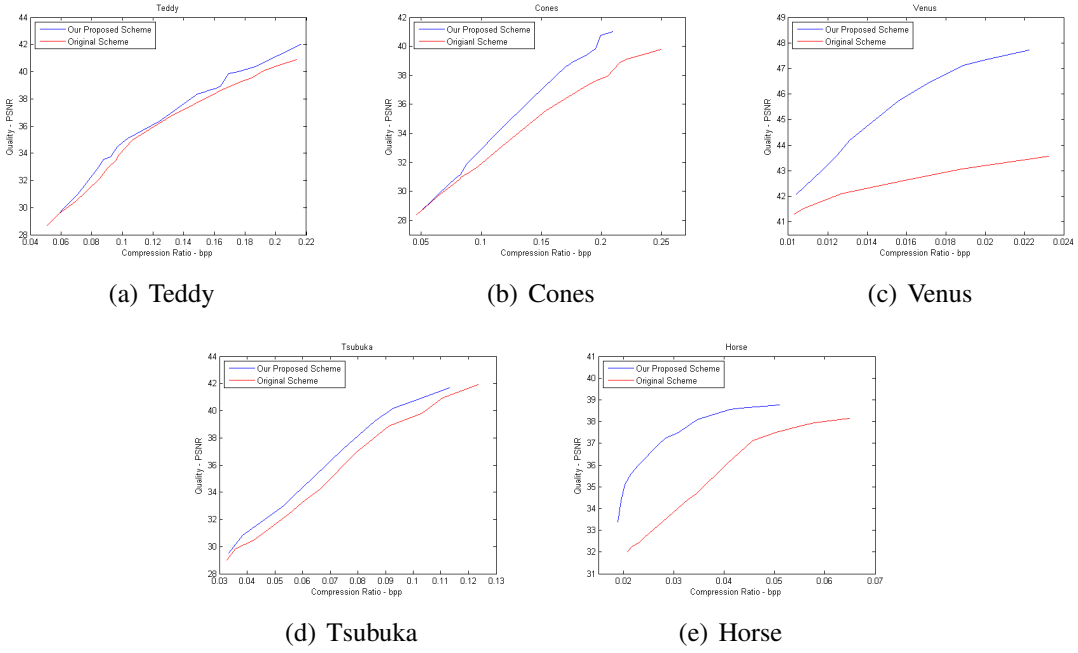
(b) Horse ground truth depth image

Figure 5.11: Depth image Tsubuka and depth image Horse.

We also tested other depth images from the database such as the "Tsubuka" image, and one computer-generated depth image the "Horse" image (Figure 5.11) [38]. The comparison

of the performance of our proposed scheme and the original scheme is shown in Figure 5.12.

Our proposed platelet based coding scheme with quadratic curve fitting technology out performs the original platelet based coding scheme (Figure 5.12). Also after comparing the reconstructed image obtained from our scheme with the reconstructed image obtained from the original paper, we also concludes that our proposed scheme reconstructed a better silhouette of the objects in the depth image. And by using the quadratic curve fitting technique we can reduce the total number of nodes generated by the quadtree decomposition method.



(a) Teddy (b) Cones (c) Venus

(d) Tsubuka (e) Horse

Figure 5.12: Comparison of Rate-Distortion performance by our proposed scheme and Morvan's original scheme.

We also record the approximate time cost of our proposed scheme and our implementation of Morvan's original scheme. The work platform we use to implement both schemes is Intel(R) Core(TM) i7 M640 CPU at 2.80 GHz and Matlab 2011b. We use the "*tictoc*" function in Matlab to record the time cost of each step. The most time consuming part of the platelet-based coding scheme is the quadtree decomposition module. The time cost of other

parts of the platelet-based coding scheme is less than 10 seconds. For different Lagrangian Multipliers, because the number of blocks generated by the decomposition module is different, the time cost of the decomposition module are also different. Generally the more number of blocks it generates, the more time cost it will be. We recorded the time cost of three depth images with different resolution (Image "Teddy" with resolution of 450×375, image "Venus" with resolution of 434×383 and image "Tsubuka" with resolution of 382×282). Then we calculate the average time cost of the decomposition module of those three images by our proposed scheme and Morvan's original scheme. Detailed time cost information can be found in Table 5.3. We can see that by using our proposed coding scheme, we can significantly reduce the time cost of the decomposition module.

Table 5.3: Comparison of time cost of the decomposition module by our proposed scheme and Morvan's original scheme

λ	Teddy(450×375)		Venus(434×383)		Tsubuka(382×282)		Average	
	Our scheme	Original scheme	Our scheme	Original scheme	Our scheme	Original scheme	Our scheme	Original scheme
10	453.23s	2374.37s	221.44s	837.01s	363.24s	1524.87s	345.97s	1578.75s
50	377.86s	2263.49s	193.46s	601.33s	269.73s	1375.29s	280.30s	1416.37s
100	337.24s	2135.38s	146.33s	586.74s	220.03s	1135.37s	234.53s	1285.83s
400	266.53s	1958.23s	105.13s	361.40s	173.86s	940.38s	181.84s	1086.67s
1100	183.94s	1729.37s	64.65s	103.58s	118.40s	883.46s	122.33s	905.47s

The experiment result also illustrates that the advantage of our proposed platelet based scheme is not only effective on a complex condition (Teddy, Cones and Tsubuka), but can be effective when the original depth image is quite simple (Venus and Horse). The experimental results met our expectations. We use less quadtree decomposition nodes and achieve a higher compression efficiency compared to the original scheme. Also when compared to the image reconstructed by Morvan's original scheme, our result has a better preservation of edge information. This improvement also enhances the quality and accuracy of the rendered 3D

image. In conclusion, our proposed platelet-based depth coding scheme successfully produce a better compression result of the original depth image.

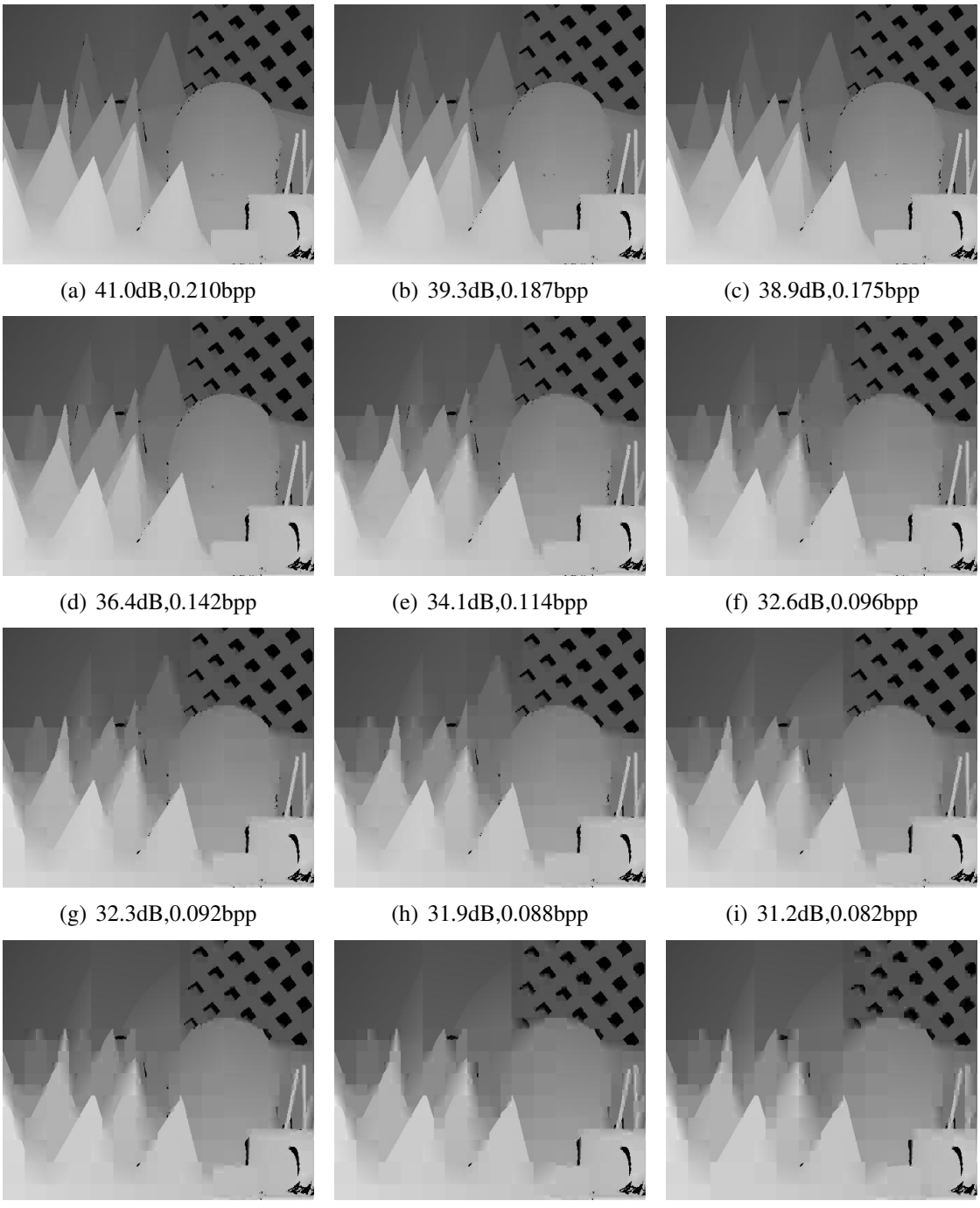
(a) 41.7dB,0.204bpp (b) 39.8dB,0.169bpp (c) 38.6dB,0.157bpp

(d) 36.3dB,0.124bpp (e) 35.1dB,0.103bpp (f) 34.6dB,0.098bpp

(g) 34.4dB,0.097bpp (h) 33.7dB,0.092bpp (i) 33.6dB,0.088bpp

(j) 32.8dB,0.084bpp (k) 31.0dB,0.070bpp (l) 29.3dB,0.056bpp

Figure 5.13: Reconstructed depth image “Teddy” at different compression rates.



(a) 41.0dB,0.210bpp (b) 39.3dB,0.187bpp (c) 38.9dB,0.175bpp

(d) 36.4dB,0.142bpp (e) 34.1dB,0.114bpp (f) 32.6dB,0.096bpp

(g) 32.3dB,0.092bpp (h) 31.9dB,0.088bpp (i) 31.2dB,0.082bpp

(j) 30.9dB,0.079bpp (k) 29.8dB,0.065bpp (l) 28.7dB,0.050bpp

Figure 5.14: Reconstructed depth image “Cones” at different compression rates.

Chapter 6

Conclusion and Future Work

3D representation technology is tending to occupy the movie market and change our traditional understanding of visual experience. It is important to achieve an accurate depth generation technique, but transmitting the depth image at the most efficient cost is also important. Compared to the traditional Multiview Video Coding scheme, directly encode the depth information could be more efficient and easy. In our thesis, a platelet coding scheme based on quadratic curve fitting technology is proposed. This scheme is based on the platelet coding scheme for depth map image firstly proposed by Morvan.

The scheme firstly proposed by Morvan is quite innovative. The platelet coding scheme take advantage of specific features of the depth image so that it keeps relatively high image quality when the compression ratio is very high. The performance of Morvan's encoder has 2 dB or 0.1 bpp improvement compared to the JPEG-2000 encoder. However, Morvan's proposed platelet coding scheme has the following three shortages. First, it cannot keep a precise silhouette of the object in the reconstructed image. Second, his implementation generates artefact errors in the reconstructed image. Finally, his proposed scheme is time consuming.

In order to improve those shortages, we proposed our platelet based coding scheme. By using quadratic curve fitting technique and changing to use a faster edge detection technique, our proposed platelet coding scheme preserves a more accurate boundary information and much less time consumption. Our proposed scheme mainly consists of three parts: the Decomposition part which is used to decompose the target image into proper sub-blocks to be modeled, the Modeling part which uses four modeling functions to obtain corresponding coefficients of each block, and the Coding & Decoding part which uses efficient codec scheme to improve the coding efficiency. Our proposed scheme further improved the quality and compression efficiency when compared to the original platelet coding scheme. Our experimental results show the performance of our proposed scheme has about 1 dB or 0.075 bpp improvement compared to the original scheme. Our proposed scheme keeps the idea and framework of the original scheme, and introduces a more effective and accurate method of fitting the silhouette of the object.

The most challenging part of implementing the platelet coding scheme is how to decide a proper factor as the judgement to generate the quadtree decomposition. Research by Ortega *et al.* [31], Fisher *et al.* [39] and Shukla *et al.* [40] prove that using Lagrangian optimization method is the most reliable factor to be the judgement in quadtree decomposition procedure. In our implementation, we control the quality and compression rate by adjusting the Lagrangian Multiplier in the Lagrangian optimization method. Also in order to reduce the number of iteration in the quadtree decomposition procedure, we choose the sprouting technique instead of the traditional pruning technique to generate the quadtree decomposition.

The experimental result shows that our proposed scheme has better performance in many aspects, such as the smaller number of nodes of the quadtree decomposition, the better quality of the reconstructed depth image and the better preservation of the boundary of the objects.

More work will be done in the future. First, in our platelet based coding scheme, we can

employ iterative algorithm to decompose every sub-block, which is quite time consuming. So we wish to implement our scheme into a parallel computation framework such as CUDA to further reduce the time consumption. Second, we expect to implement our image based work into video coding schemes. Third, right now our proposed scheme does not have the ability to fast decompose a high definition image (resolution of 1280×720 and more). By improving the efficiency of our proposed scheme in the future, we hope our scheme could decompose high definition image faster Finally, in order to further improve the compression performance of the encoder, we expect to investigate the merging technique [41] to further reduce the number of blocks needs to be approximated.

References

[1] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3DTV – A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1606–1621, November 2007.

[2] P. Merkle, K.Mueller, A. Smolic, and T. Wiegand, "Statistical evaluation of spatiotemporal prediction for multiview video coding," in *Proceeding of Workshop on Immersive Communication and Broadcast Systems*, Berlin, Germany, October 2005, pp. 1–6.

[3] A. Kaup and U. Fecker, "Analysis of multireference block matching for multiview video coding," in *Proceedings of 7th Workshop Digital Broadcasting*, Erlangen, Germany, September 2006, pp. 33–39.

[4] C. Fehn, P. Kauff, M. O. de Beeck, F. Ernst, W. Jsselsteijn, M. Pollefeys, L. V. Gool, E. Ofek, and I. Sexton, "An evolutionary and optimised approach on 3D-TV," in *Proceedings of International Broadcast Conference*, Amsterdam, Netherlands, September 2002, pp. 357–365.

[5] C. Fehn, "3D-TV using depth-image-based rendering (DIBR)," in *Proceedings of Picture Coding Symposium (PCS'04)*, San Fransisco, USA, December 2004, pp. 1–6.

[6] Y. Morvan, D. Farin, and P. H. N. de With, "Novel coding technique for depth images using quadtree decomposition and plane approximation," in *Proceedings of the International Society for optics and photonics*, vol. 5960, July 2005, pp. 1–8.

[7] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *Proceedings of the International Society for optics and photonics*, vol. 6055, Singapore, January 2006, pp. 1–10.

[8] F. Guo, Y. He, N. Sakr, J. Zhao, and A. E. Saddik, "Haptic data compression based on quadratic curve reconstruction and prediction," in *Proceedings of the Third International Conference on Internet Multimedia Computer and Service (ICIMCS '11)*, Chengdu, China, August 2011, pp. 193–196.

[9] H. Kalva, L. Christodoulou, L. M. Mayron, O. Marques, and B. Furht, "Design and evaluation of a 3D video system based on H.264 view coding," in *Proceedings of the 2006 International workshop on Network and operating systems support for digital audio and video (NOSSDAV' 06)*, no. 12, New York, USA, May 2006, pp. 1–8.

[10] H. Kalva, L. Christodoulou, L. Mayron, O. Marques, and B. Furht, "Challenges and opportunities in video coding for 3DTV," in *IEEE International Conference on Multimedia & Expo (ICME)*, Toronto, Canada, July 2006, pp. 1689–1692.

[11] C. Bilen, A. Aksay, and G. B. Akar, "A Multi-View Video Codec based on H.264," in *2006 IEEE International Conference on Image Processing (ICIP' 2006)*, Atlanta, USA, October 2006, pp. 541–544.

[12] D. Socek, D. Culibrk, H. Kalva, O. Marques, and B. Furht, "Permutation-based low-complexity alternate coding in multiview H.264/AVC," in *IEEE International Conference on Multimedia & Expo (ICME)*, Toronto, Canada, July 2006, pp. 2141–2144.

[13] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierachical B-pictures and MCT-F," in *IEEE International Conference on Multimedia & Expo (ICME)*, Toronto, Canada, July 2006, pp. 1929–1932.

[14] K.-J. Oh and Y.-S. Ho, "Multi-view video coding based on the lattice-like pyramid GOP structure," in *Proceedings of Picture Coding Symposium (PCS' 06)*, Beijing, China, April 2006, pp. 1–6.

[15] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Efficient prediction structures for multi-view video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461–1473, November 2007.

[16] X. Li, D. Zhao, S. Ma, and W. Gao, "Fast disparity and motion estimation based on correlations for multiview video coding," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 2037–2044, November 2008.

[17] L. Shen, Z. Liu, T. Yan, Z. Zhang, and P. An, "View-adaptive motion estimation and disparity estimation for low complexity multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 925–930, January 2010.

[18] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable Multiview Video Coding using 3-D warping with depth map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1485–1495, November 2007.

[19] J.-U. Garbas and A. Kaup, "Enhancing coding efficiency in spatial scalable Multiview Video Coding with wavelets," in *2009 IEEE International Workshop on Multimedia Signal Processing (MMSP' 09)*, Rio De Janeiro, Brazil, October 2009, pp. 1–6.

[20] J. Zhang, M. M. Hannuksela, and H. Li, "Joint multiview video plus depth coding," in *2010 IEEE 17th International Conference on Image Processing (ICIP' 2010)*, Hong Kong, September 2010, pp. 2865–2868.

[21] K. Mueller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, April 2011.

[22] R. Krishnamurthy, B. Chai, H. Tao, and S. Sethuraman, "Compression and transmission of depth maps for image-based rendering," in *2001 IEEE International Conference on Image Processing (ICIP' 2001)*, vol. 3, Thessaloniki, October 2001, pp. 828–831.

[23] ATTEST.project, http://www.hitech-projects.com/euprojects/attest/index.htm, Last visit on Sep 16, 2012.

[24] Y.-C. Fan, S.-F. Wu, and B.-L. Lin, "Three-dimensional depth map motion estimation and compensation for 3D video compression," *IEEE Transactions on Magnetics*, vol. 47, no. 3, pp. 691–695, March 2011.

[25] Y. Liu, Q. Huang, S. Ma, D. Zhao, W. Gao, S. Ci, and H. Tang, "A novel rate control technique for multiview video plus depth based 3D video coding," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 562–571, June 2011.

[26] P. Zanuttigh and G. M. Cortelazzo, "Compression of depth information for 3D rendering," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, Potsdam, May 2009, pp. 1–4.

[27] D. Graziosi, N. Rodrigues, C. Pagliari, S. de Faria, E. da Silva, and M. de Carvalho, "Compressing depth maps using multiscale recurrent pattern image coding," *Electronics Letters*, vol. 46, no. 5, pp. 340–341, March 2010.

[28] D. Sebai, F. Chaieb, K. Mamou, and F. Ghorbel, "Piecewise linear function estimation for platelet-based depth maps coding using edge detection," *Three-Dimensional Image Processing (3DIP) and Applications II, Proceeding of SPIE-IS&T Electrical Imaging*, vol. 8290, pp. 1–6, 2012.

[29] M. Sarkis and K. Diepold, "Depth map compression via compressed sensing," in *16th IEEE International Conference on Image Processing (ICIP), 2009*, Cairo, November 2009, pp. 737–740.

[30] D. D. Silva, W. Fernando, H. Kodikaraarachchi, S. Worrall, and A. Kondoz, "A depth map post-processing technique for 3D-TV systems based on compression artifact analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 99, pp. 1–30, August 2011.

[31] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 1, no. 6, pp. 23–50, November 1998.

[32] G. M. Schuster and A. K. Katsaggelos, "The minimum-average and minimum-maximum criteria in lossy compression," *Vistas in Astronomy*, vol. 41, no. 3, pp. 427–437, September 1997.

[33] G. Schuster and A. K. Katsaggelos, "The min-max approach in video coding," in *Proceeding of 1997 International Conference on Acoustics Speech and Signal Proceesing (ICASSP'97)*, vol. IV, Munich, Germany, April 1997, pp. 3105–3018.

[34] D. Hoang, "Fast and efficient algorithms for text and video compressoin," Ph.D. dissertation, Brown University, 1997.

[35] H. Everett, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, May 1963.

[36] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 299–315, March 1989.

[37] vision.middlebury.edu, http://vision.middlebury.edu/stereo, Last visit on August 31, 2012.

[38] CS766 Project 3: Photometric Stereo, http://pages.cs.wisc.edu/lizhang/courses/cs766-2008f/projects/phs/students/barnard/report.html, Last visit on Oct 6, 2012, Course Project.

[39] M. L. Fisher, "The Lagrangian relaxation method for solving interger programming problems," *Management Science*, vol. 27, no. 1, pp. 1–18, January 1981.

[40] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree structured compression algorithms for piecewise polynomial images," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 343–359, March 2005.

[41] S. Oudin, P. Helle, J. Stegemann, C. Bartnik, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Block merging for quadtree-based video coding," in *2011 IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, July 2011, pp. 1–6.