Hiding Depth Map in JPEG Image and MPEG-2 Video

by

Wenyi 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 2011

Abstract

Digital watermarking of multimedia content has been proposed as a method for different applications such as copyright protection, content authentication, transaction tracking and data hiding.

In this thesis, we propose a lossless watermarking approach based on Discrete Cosine Transform (DCT) for a new application of watermarking. A depth map obtained from a stereoscopic image pair is embedded into one of the two images using a reversible watermarking algorithm. Different from existing approaches which hide depth map in spatial domain, the depth information is hidden in the quantized DCT domain of the stereo image in our method. This modification makes the watermarking algorithm compatible with JPEG and MPEG-2 compression.

After the investigation of the quantized DCT coefficients distribution of the compressed image and video, The bit-shift operation is utilized to embed the depth map into its associated 2D image reversibly for the purpose of achieving high compression efficiency of the watermarked image and/or video and high visual quality of stereo image and/or video after the depth map is extracted.

We implement the proposed method to analyze its performance. The experimental results show that a very high payload of watermark (e.g. depth map) can be embedded into the JPEG compressed image and MPEG-2 video. The compression efficiency is only slightly reduced after the watermark embedding and the quality of the original image or video can be restored completely at the decoder side.

Contents

List of Tables

List of Acronyms

BS	Bit Shift
IBR	Image Based Rendering
LSB	Least Significant Bit
QIM	Quantization Index Modulation
HVS	Human Visual System
DCT	Discrete Cosine Transform
PVD	Pixel Value Difference
BER	Bit Error Rate
DWT	Discrete Wavelet Transform
PSNR	Peak Signal to Noise Ratio
JPEG	Joint Photographic Experts Group
MPEG	Moving Pictures Experts Group

This thesis is dedicated to my family.

Acknowledgement

It is a great pleasure to thank those who helped to make this thesis possible.

I would like to owe my deepest gratitude to my supervisor, Professor Jiying Zhao, whose encouragement, patience and guidance enabled me to understand and deal with my project during every step of work.

I am grateful to offer my regards and blessings to all of my colleagues in our lab who supported me during my study here.

Chapter 1

Introduction

1.1 Stereo vision

Extending the visual experience from 2D scenario to the third dimension has been compelling for decades and it is surprising that the three dimensional imaging has almost two hundred years of investigation history [1]. In 1838, Wheatstone introduced the first stereoscope device which gave different view images to the left and right eye simultaneously and separately. And the stereoscopic visual effect was widely applied in theaters with the improvement of stereo imaging after 1930 [2]. In the stereo view system, two

Figure 1.1: Early stereo image pair.

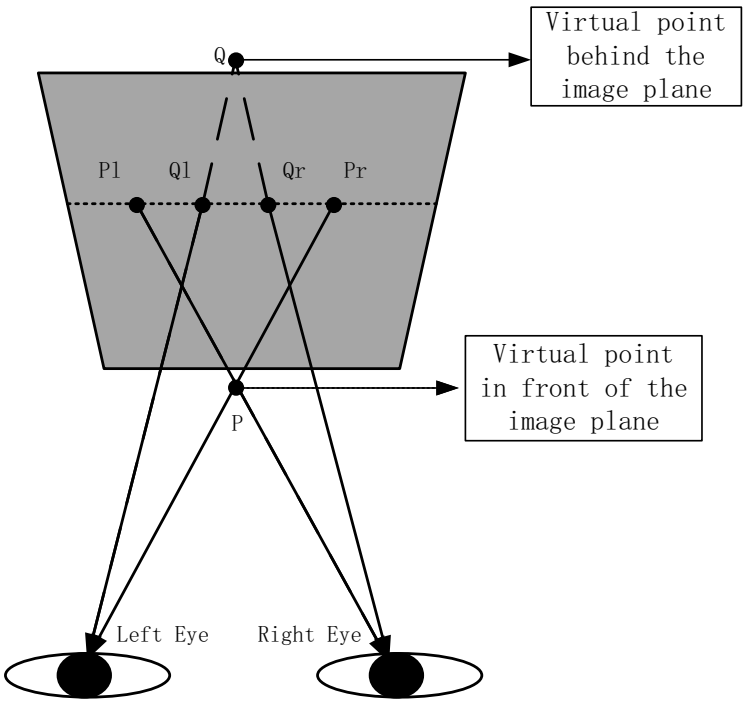


Figure 1.2: Depth representation from two disparity views.

images which are taken under two slightly different view points are presented separately to each eye of human and the difference of the view point can simulate the experience of the natural performance of human eyes. In these two stereo images, two regions, which stand for the same object, are shown to each eye separately. The slightly position difference of these two regions would result in a parallax for human sense. In this case, the object position for human sense is out of the the image plane and the depth sense will depend on the disparity of these two corresponding regions.

Other than using two stereo images to simulate the sense of 3D directly, the researchers also interested in the data information about the distance between the object and the cameras. In this case, the extraction of depth information is researched for decades. The depth information are presentations which can be captured and computed from the real world with cameras or other devices. In a depth map, a visibility-limited

model (gray scale image) of the scene is presented, in which the shape of each object in different positions are colored in different gray scale with regard to the distance between the object and the camera.

The methods for depth information capture can be classified into two categories: active depth sensor methods and passive depth calculation methods. For active depth sensor methods, the optical sensors, such as laser sensor, infrared ray sensor, or light pattern sensor, are used to obtain the depth information directly. On the other hand, the depth map are generated from images captured by cameras indirectly in passive depth calculation methods such as stereo matching or shape from motion. The cost of passive methods are lower than the active methods but the accuracy of passive methods is also lower compared to active ones.

Since the active methods are difficult to be implemented, most of the methods presenting the 3D sense use the passive way which depends on disparity in stereo view or multi-view systems.

The extraction of the depth information, which can be used to reconstruct the 3D scene in the later process, and the reconstruction of the 3D scene in machine vision technology is demonstrated in Fig. 1.3.

With regard to the extraction of depth map from the disparity map, the triangulation is used in Fig. 1.4. The triangulation need to be achieved by only the information of the stereo image pair and the calibrated cameras. For the information of stereo images, the location difference of the same region in each stereo image is needed to be determined. After the location difference which is called as disparity is determined, the depth information can be calculated by triangulation.

In Fig. 1.4: x (meter) and z (meter) are the coordinates in 3D space, x (pixel) is the coordinate in 2D image. B is the length of baseline which is the distance between focal

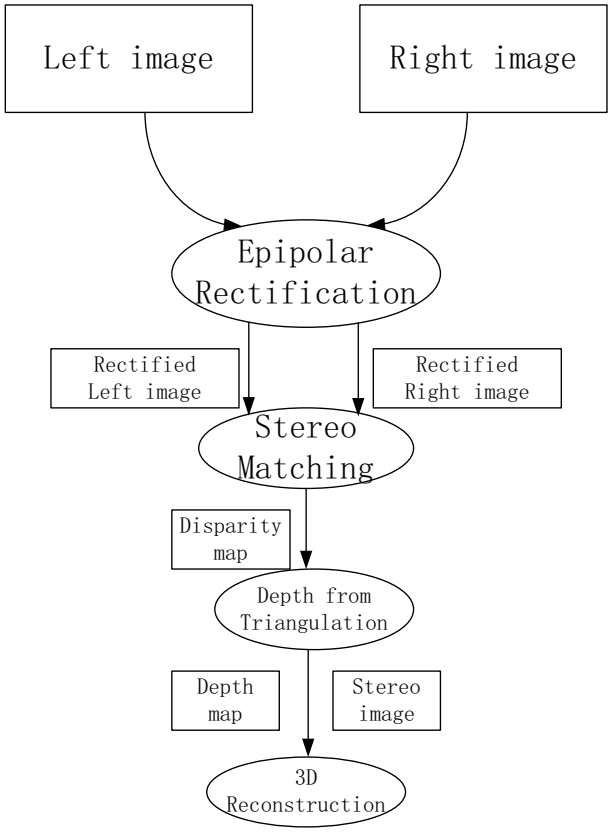


Figure 1.3: Stereo flowchart.

points of these two cameras. Points C_l and C_r are the focal points of two cameras. f is the focal length, all of these parameters are already known from the camera itself or the cameras' position in the 3D space; X_l and X_r are the x coordinates of the corresponding points in the stereo images, these values can be got from stereo matching algorithms; Z is the depth of the point in 3D space, this value is unknown and it is also what we want to calculate.

From similar triangular, we can get:

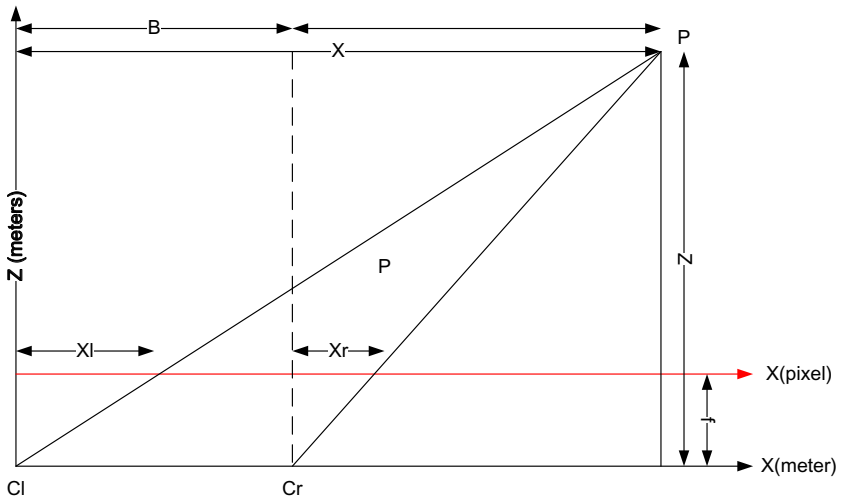


Figure 1.4: Depth from triangulation.

$$\frac{X}{Z} = \frac{X_i}{f} \tag{1.1}$$

$$\frac{X - B}{Z} = \frac{X_r}{f} \tag{1.2}$$

Then we can get

$$X = \frac{Z \cdot X_i}{f} \tag{1.3}$$

$$X = \frac{Z \cdot X_r}{f} + B \tag{1.4}$$

After combining these two equations:

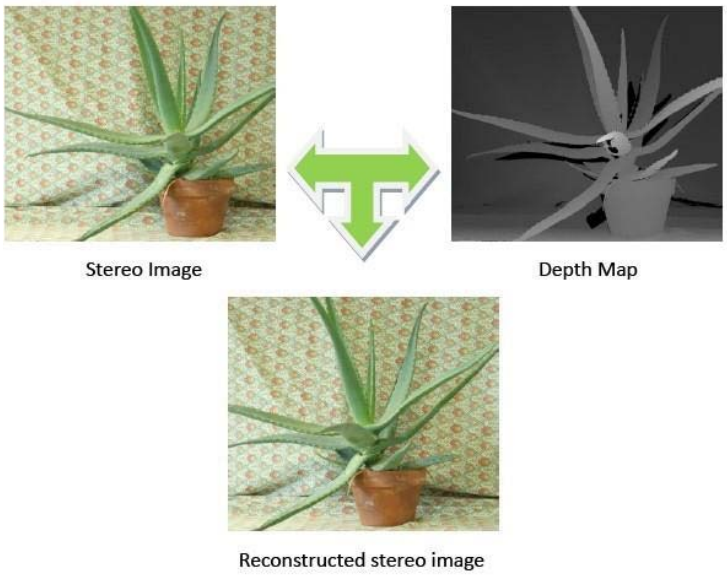


Figure 1.5: Stereo image generation.

$$\frac{Z \cdot X_r}{f} + B = \frac{Z \cdot X_i}{f} \tag{1.5}$$

$$\begin{aligned} Z &= \frac{Bf}{Xl - Xr} \\ &= \frac{Bf}{d} \end{aligned} \tag{1.6}$$

where d is the disparity value and we can now get the depth information from two stereo images.

With the depth map and one of the stereo image, we can regenerate a series of stereo images with the reference stereo image. After obtaining these stereo images, the viewers can experience the 3D vision.

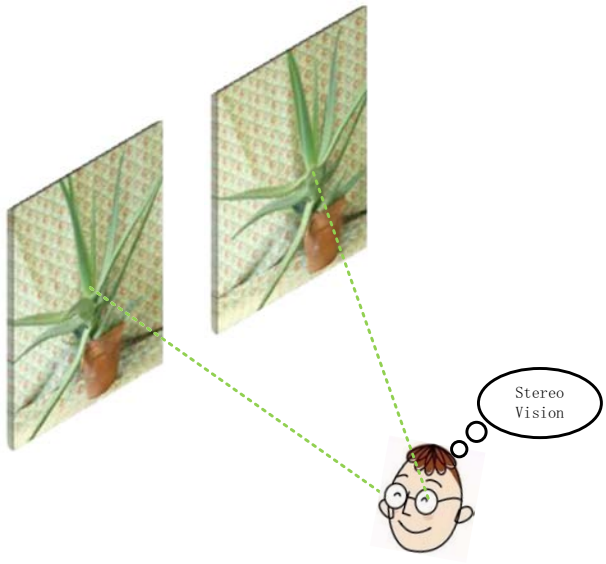


Figure 1.6: 3D experience.

1.2 Digital watermarking

1.2.1 Information hiding

The idea of secrete communication is almost as old as communications itself. The art of steganography can be found in ancient time during which sympathetic inks were used to write invisible. Watermarking methods are evolved from steganography. The early watermarking technique is to impress the paper a form of image or text. Although steganography and watermarking are both techniques which are used to hide the information into an cover work, they are still two different conceptions.

The steganography and watermarking can be differed by the intent of usage. The watermark can be perceived as side information of the cover content. It can be information about copyright, license, authorship and etc. On the other hand, the embedded information of steganography may have nothing to do with the cover content.

1.2.2 Digital watermarking scheme

In the modern world, the rapid development of computer internet and information technology has explored means of new business and entertainments. In all of these new applications, the digital data is used indispensable. And besides, the digital information can be duplicated and distributed easily from internet and mobile disks. In this case, the effective copyright protection methods are required and the concept of digital watermarking came up to deal with the problems related to the protection and identity of media sources.

A typical watermarking system (Fig. 1.7) consists of watermark embedding and watermark extraction.

The inputs of the embedding process are watermark, cover media data and embedding security key. The watermark is always a sequence of binary bits. The cover media data is the the watermark information carrier in which the watermark bit sequence are hiding into it invisibly or visibly. The key is used to guarantee the authorized access of the watermark information and therefore the security of the watermarking system can be improved. The output of the watermark embedding process is the watermarked data.

For the watermark extraction process, the inputs are the watermarked data, the security key and ,depending on the watermarking scheme, the original cover and/or the original watermark. The output is the recovered watermark and ,also depending on the watermarking scheme, the recovered carrier data. Supposed that a watermark message is defined as M, C is the carrier message and K is the security key. In watermarking scheme, an embedding function $e(.)$ takes the watermark M, the carrier data C and the security K as input messages and outputs the watermarked data C'.

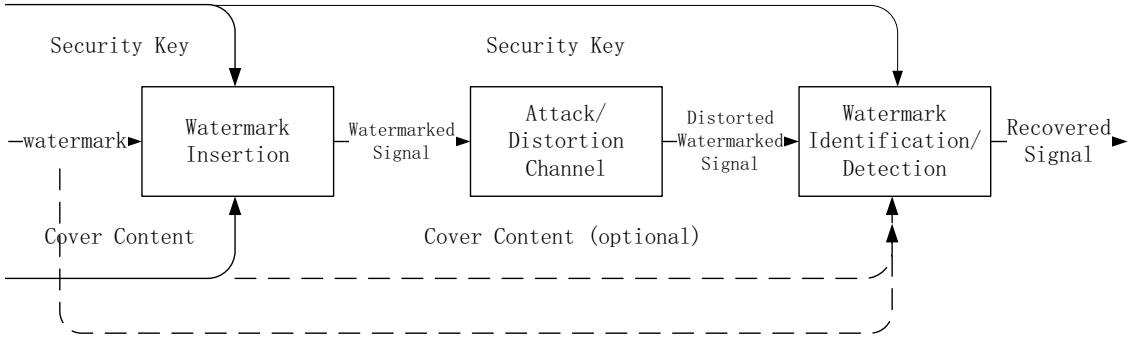


Figure 1.7: Watermarking system.

$$C' = e(C, M, K) \tag{1.7}$$

At the receiver side, the extraction procedure is depicted in the following equation:

$$M' = d(C', K, ...) \tag{1.8}$$

where$d(.)$ is the detector function. C and M are optional inputs for the detector function depending on the watermarking scheme.

With the general embedding and extraction procedures, the digital watermarking can be further classified into different categories in Fig. 1.8.

According to the embedding domain of the cover content, the watermarking algorithms can be classified into two groups: spatial domain embedding and frequency domain embedding. If the watermark is embedded into the spatial domain, the magni-

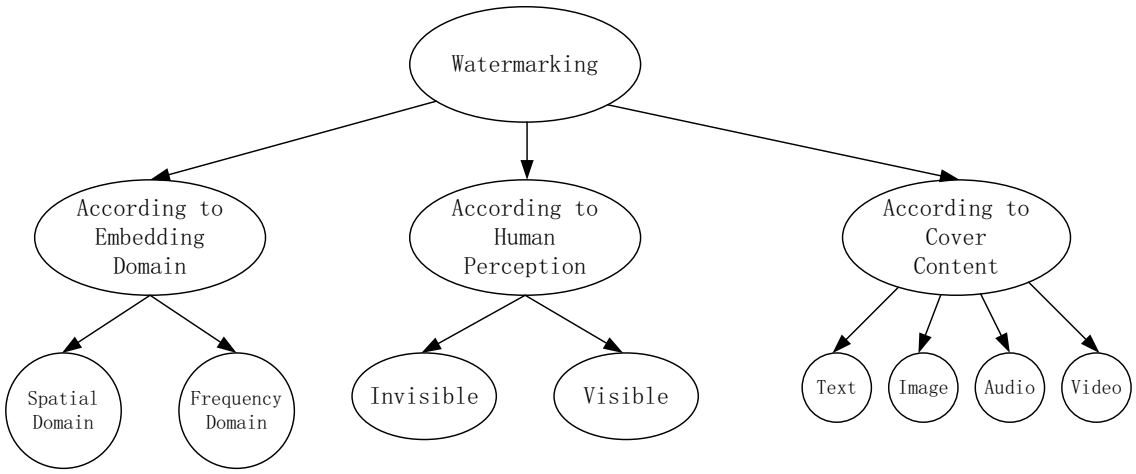


Figure 1.8: Watermarking categories.

tude of the cover signal are modified directly. If the watermark is embedded into the frequency domain, the digital signal are pre-transformed by DCT or DWT, and then the watermark is embedded by changing the magnitude of each frequency component.

According to the human perception, the watermarking scheme can be considered as invisible watermarking and visible watermarking. For invisible watermarking, the watermark is added by slightly changing the cover content and the HVS are always considered to prevent any noticeable difference between the original cover content and the watermarked content. If the watermark is embedded visibly, it is often some noticeable logo which declares the authorship of the digital content which prevents the illegal access of the original cover content.

According to the cover content, the watermarking can be classified into text watermarking, image watermarking, audio watermarking and video watermarking. With the development of internet, there are so many images on the World Wide Web without any authorize protection. And besides, most of the anti-compression image watermarking can be applied to video watermarking since that video are sequence of images com-

pressed together. In this case, most of the current research is based on the image watermarking.

In this thesis, most of the work is focused on the image watermarking, and the scheme is also applied to the video watermarking.

1.2.3 Digital image watermarking

Various techniques have been proposed for the watermarking of still image data. The watermark embedding scheme are designed to either insert the watermark directly into the original data such as the color components or into some transformed domain of the original data to take advantage of perceptual properties or robustness to particular signals attacks.

The required characteristics for image watermarking include invisibility, robustness, and capacity.

1. Invisibility: In order to minimize the noticeable distortion to the cover image, the perceptual models are often used to scale the watermark signal or determine the appropriate embedding location in the cover image.

2. Robustness: The watermark information should survive after certain common signal processing operations and attacks. These processing includes compression such as JPEG, filtering, rotation, cropping, scaling, A/D and D/A conversion, and additive noise.

3. Capacity: the capacity refers to the amount of information which can be inserted into the cover image and extracted reliably afterwards.

A good watermarking scheme should achieve a good trade-off among these requirements, refer to Fig. 1.9.

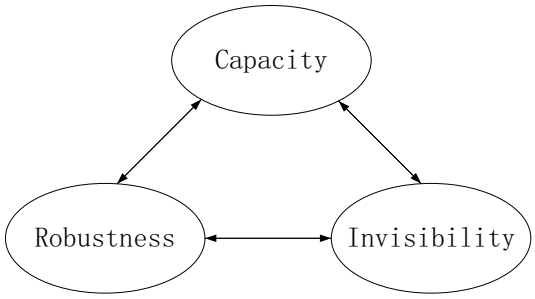


Figure 1.9: Watermarking characteristics.

1.2.4 The applications of digital watermarking

Digital watermarking can be applied to a wide rang of applications:

1. Copyright protection and authentication. The copyright protection is one of the main reasons to propose the watermarking scheme. The idea is to hide the information about the ownership into the cover image in order to prevent other people from illegally declaring the property right of the image content.

2. Source tracking. This category of watermarking is used to track the distribution of the cover image. A watermark is embedded into a digital signal at each point of distribution, which means that different recipients can get different watermarked content. If a illegal copy of the work is found later, then the watermark may be retrieved from the copy and the source of the illegal distribution is known.

3. covert communication. In this category of watermarking application, the watermark information can be some secret message which are only accessible to authorized people. At the receiver side, only the people knowing the watermarking embedding scheme and the security key can obtain the hiding information.

1.3 The contributions of the thesis

In this thesis, we propose a lossless watermarking approach based on Discrete Cosine Transform (DCT) for a new application of watermarking. A depth map obtained from a stereoscopic image pair is embedded into one of the two images using a reversible watermarking algorithm. Different from existing approaches which hide depth map in spatial domain, the depth information is hidden in the quantized DCT domain of the stereo image in our method. This modification makes the watermarking algorithm compatible with JPEG and MPEG-2 compression.

After the investigation of the quantized DCT coefficients distribution of the compressed image and video, The bit-shift operation is utilized to embed the depth map into its associated 2D image reversibly for the purpose of achieving high compression efficiency of the watermarked multimedia content and high visual quality stereo image after the depth map is extracted.

We implement the proposed method to analyze its performance. The experimental results show that a very high payload of watermark (e.g. depth map) can be embedded into the JPEG compressed image and MPEG-2 video. The compression efficiency is only slightly reduced after the watermark embedding and the quality of the original image or video can be restored completely at the decoder side.

Publications generated from the research:

[1] Wenyi Wang, Jiying Zhao, James Tam, Filippo Speranza, and Zhou Wang, Hiding Depth Map into Stereo Image in JPEG Format using Reversible Watermarking, ICIMCS'11, pp. 82-85, August 5-7, 2011, Chengdu, Sichuan, China.

1.4 Thesis structure

In Chapter 2, we give an introduction to existing approaches for depth map watermarking, robust watermarking for JPEG compression and reversible watermarking, in Chapter 3 we propose our scheme with describing the watermarking embedding and extraction, in Chapter 4.1 we illustrate and evaluate the proposed scheme and in Chapter 5 we conclude the thesis and give some suggestions and ideas for future research work.

Chapter 2

Literature review

2.1 Existing depth map watermarking approaches

Stereo vision, a naturally performed human body function, has been recognized as being compelling to further improve the machine vision technology. It is obviously that a stereo pair of images should be analyzed, transmitted and stored to access the the stereo information. In this case, the reduction of the transmission bandwidth and storage requirement is a considerable concern when we operate the stereo images comparing the conventional image. A straightforward approach of this problem is to simply compress and encode the stereo media pair separately and the storage requirement are nearly twice as much as the requirement of the single content [6]. An alternative approach utilizes the disparity estimation and disparity compensation schemes. The basic idea of this approach is to use the stereo media pair to generate a disparity map or a depth map which can be used to estimate the other stereo image with regarding to only one of stereo image pair. In this case, only the disparity map, one of the stereo image and sometimes the estimation error are encoded, transmitted and stored [7].

Some researchers proposed a different approach which not only half the storage requirement and transmission bandwidth but also control the access of depth information. In this approach, the basic idea is to embed the depth information, which can recover the other stereo image, into the the reference image using watermarking algorithm. Thus, only one stereo image containing the depth information should be stored or transmitted in stead of two stereo images or one stereo image plus the depth map. And at the receiver side, the depth map can be extracted and be used to regenerate the other stereo image.

D. Coltuc propose a method to embedding the depth information into one of the stereo image by digital watermarking using integer transform and mod operation [8][9][10]. In his paper, the six test color images are from [11]. The watermark in the color images is segmented into each of three color plane using reversible watermarking. The watermark capacity is generally around 2.5 bpp per color channel and 6 bpp per color pixel. After applying JPEG compression to the depth information and estimation error, the required watermark capacity can be satisfied. And the host image can be recovered at the receiver side after the watermark is extracted.

Different from D. Coltuc using the dense disparity map as watermark, J.N. Ellinas used a block based disparity compensation to generate a sparse disparity map as watermark to be embedded into the stereo image. There are three test grey scale image in his paper and the watermark capacity is around 1.4 bpp per pixel. After applying JPEG compression to the disparity information with a very high quality factor (96 for image 'room', 85 for image 'SYN.256', 92 for image 'fruit'), the disparity map were embedded successfully using reversible watermarking method PVD (pixel value difference) combined with HVS(human visual system) [12].

A. Khan and his colleagues embedded the depth information in another way. In their

approach, the depth information was embedded into the HL and LH bandwidth of the DWT domain of the original image using variable threshold [13][14]. The watermarking embedding method is also applied reversible watermarking of bit shifting. The capacity of this embedding method can achieve 0.75 bpp.

We noticed that although the previous work performed very well with regarding the watermarking capacity, watermark invisibility, and original image reconstruction, all of the approaches are fragile watermarking methods and the watermark information can not survive the JPEG compression. This is because that the DCT coefficients in JPEG images are multiples of the corresponding quantization factors. If a watermark is added into a JPEG compressed image, the distortion to the coefficients are usually small compare to the quantization parameter, and this small distortion are very likely to be removed after the quantization. In this case, although the size of the stereo image pair can be halved by hiding the depth information into one of the stereo images, the image size can not be reduced further using lossy compression methods which are widely used in image or video storage or transmission.

2.2 Conventional anti-compression watermarking algorithms

To investigate whether it is possible to hiding the depth information using anti-compresssion watermarking method, the watermark capacity must be examined first.

High capacity watermarking embedding is very challenging for the reason that it has to satisfy the characteristics of in-visibility and robust to compression or other signal processing, which are both conflicts to watermarking capacity.

As far as how to obtain a robust watermarking embedding method is concerned,

the block based embedding is widely used due to its robustness to attacks such as image compression. Swanson et al. [15][16] were the first researchers who utilized the block based information hiding scheme using the projection technique. This kind of method can be understood as an energy spreading process. The blocks are vectored and projected onto random vectors with regarding to quantization step and the embedding bit. In other words, a base vector (normally a random generated vector) are spread over a block to denote one watermark bit. In this case, there is redundancy information of one watermark bit embedded into the host. After the compression or other attacks, we can still extract the watermark bit correctly from calculating the correlation between the watermarked image and the base vector.

L.Tse-Hua et al. [17] proposed a robust watermarking method which performed very well in both capacity and anti-compression. And in the anti-compression watermarking schemes, it has the top capacity. This algorithm is based on the quantized projection embedding algorithm in which a random permutation of the columns of a Hadamard matrix as projection vectors and a fixed perceptual mask based on the JPEG compression quantization matrix. As a result, the embedding capacity can reach as high as 0.09 bpp.

Recently, N.K. Kalantari and S.M. Ahadi [18] proposed a novel arrangement for quantizer based on the Quantization Index Modulation (QIM). In their method, they achieved a better BER under the same compression ratio compared to T. Lan's method but the watermark capacity, which is about 0.05 bpp in the experimental result of the paper, is not mentioned to be improved.

However, the watermark capacity of the previous anti-compresstion watermarking schemes is much less than the capacity of the watermarking schemes which are used to hide the depth information in Section 2.1. In this case, we have to further investigate

the possible depth map size after compression.

2.3 Depth map compression

Since the image based rendering (IBR) methods utilize depth map and the stereo or multi-view images or videos to reconstruct the 3D sense [19], the particular encoding method for depth map is researched widely recently.

In general, the compression of the depth information is processed by using the standard compression algorithm such as JPEG 2000 or H.264 with special technical to preserve the sharp edge information. This is because that the edge information is usually associated with the high frequency of the depth map and these information would be remove significantly by those compression standards.

In [20], a region of interest is used to prevent JPEG2000 compression artifacts. This is rest on a assumption that the depth maps are usually consist of a group of smooth regions and some sharp edges between those flat regions. The sharp regions are the key issue in the depth map compression because even some small errors may cause severe effect to the further applications of depth map such as the generation of stereo images. In this approach, the depth map is segmented and the edge information is predicted from the difference between the original depth map and the segmented depth map. Afterwards, the difference is lossless compressed and the segmented information is lossy compressed at a higher ratio. As a result, the bit rate of the compressed depth map is around 0.2 bpp with a PSNR value near 50 dB.

In [21], the filtering techniques are used to further improve the compression performance of an H.264 encoder. The authors also mentioned that the depth map is spatially monotonous except the sharp edges. In this case the compression distortion are always

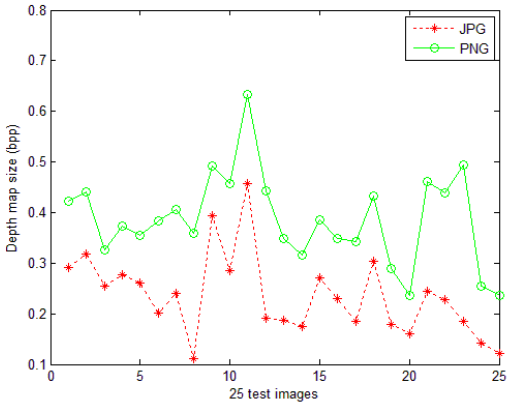


Figure 2.1: The depth map size in JPEG and PNG format.

concentrated at the object boundaries in the reason that the edge information is in high frequency which are likely to be removed in general compression scheme. In this approach, the frequent-close filter is utilized to emphasize the boundary information and this can benefit the compression result of depth map. As a result, the bit rate of the compressed depth map is around 0.5 bpp with a PSNR value about 47 dB.

In [27], another method called multidimensional multiscale parser is utilized and it performed the best up to 2010. In this approach, the depth map can be compressed to 0.129 bpp while obtain a PSNR value of 50.05 dB.

More generally, we also tested 29 depth maps using JPEG compression with a PSNR value around 30 dB and PNG lossless encoding to examine the size of a depth map in ordinary lossy and lossless compression format.

From Fig. 2.1, it can be observed that, the size of JPEG compressed image varies from 0.1 bpp to 0.45 bpp, the size of PNG image varies from 0.21 bpp to 0.65 bpp. The reason for applying the lossless compression (PNG format) is that there are some situations that a precise depth information is required such as medical use and military

use. For color images, we can hide the depth information into each of the three channels so that the watermarking capacity for each channel is only one third of the whole capacity requirement. This means that the watermarking capacity requirement for each channel is 0.15 bpp for JPEG compressed depth map and 0.22 bpp for depth map in PNG format.

In Section 2.2, it can be observed that the watermarking capacity of conventional anti-compression methods is always below 0.1 bpp and it is not enough to hide the depth information. And besides, all of the existing works of hiding depth map into stereo images used the reversible watermarking algorithm in their approaches. This provokes us to study the reversible watermarking algorithm to find the possible solution to hide the depth map into a JPEG compressed image.

2.4 Reversible watermarking

From the previous section, we get the required watermarking capacity to be either 0.15 bpp or 0.22 bpp depending on the compression method applied to the depth map. This requirement of watermarking capacity is seldom achieved by conventional watermarking methods which are robust to JPEG compression. In Lan's paper [17], an improved quantized projection based embedding method is proposed with the characteristic of anti-compression. The capacity of their scheme is up to 0.09 bpp which outperforms most of the anti-compression watermarking algorithm. However a watermarking method with the capacity of 0.09 bpp is still not enough for the requirement. In this case, we turn to the fragile watermarking method with high capacity and high visual quality to meet the requirement of high capacity.

The target of obtaining both the high visual quality and high capacity is another

challenge because they are two conflict requirements for watermarking scheme. When we embed lots of information into the original image, it is obvious that there will be a huge change to the original image which leads to a serious visual quality damage. To solve this problem, we focus on a special watermarking method, called reversible watermark. A reversible watermarking scheme means that the original image can be completely restored at the receiver side after the watermark is extracted. In this case, no matter how much information is embedded into the original image and how severe damage is introduced by it, we can always completely remove the damage and recover the original image with good quality.

After years of development in the field of reversible watermarking, this kind of watermarking can be classified into three categories[28]:

1. Schemes that compress part of the original image to release the space for the watermark information [29]-[31].

2. Schemes that apply difference expansion to the original image and bit-shift the calculated difference value to release the space for the watermark information [32]-[42].

3. Schemes that exchange the histogram bins of blocks in the original image to represent the watermark information [43]-[45].

In the following part, the basic ideas of these three schemes are introduced.

2.4.1 Reversible watermarking by compressing part of the original image

When we embed watermark into the original image, some part of the image will be replaced by the watermark such as the least significant bits (LSB). In order to recover the original image, one straightforward way is to compress these replaced bits and append them to the watermark information to generate a new watermark bit stream and then embed the watermark with the recovering information by replacing the compressed part in the original image. The basic idea from Celik et al. [29] is presented to demonstrate the general scheme of this kind of reversible watermarking.

Assuming a vector containing N pixel values is $x = (x_0, x_1, x_2, \cdots, x_N)$, this vector is quantized by an integer factor L by applying:

$$v_i^{\cdot} = \left\lfloor \frac{x_i}{L} \right\rfloor \tag{2.1}$$

where $\lfloor x \rfloor$ denotes the largest integer no larger than x.

After calculating the quantized value, the error introduced by quantization is calculated:

$$e_i = x_i - L \cdot x_i^{\cdot} \tag{2.2}$$

Now, we have a vector of quantization error $e = (e_0, e_1, e_2, \cdots, e_N)$. By applying some lossless compression algorithm, we can regenerate vector e to e_c= $(e_{0c}, e_{1c}, e_{2c}, \cdots, e_{Nc})$, where $N_c < N$. In this case, we release a vacancy of $N - N_c$ for values ranging from $0 - (N-1)$. After transform the binary values of watermark to the required values, we can get $w = (w_0, w_1, w_2, \cdots, w_{N-N_c})$. By appending the water-

Figure 2.2: The original 8-bit value.

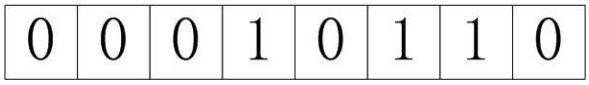


Figure 2.3: The 1 bit shifted 8-bit value.

mark w to the compressed error e_c, we get the total information we need to embed into the original $w_0 = (e_{0c}, e_{1c}, e_{2c}, \cdots, e_{N_c}, w_0, w_1 \cdots w_{N-N_c})$. The last step is to embed the watermark w_0 to the original image by applying:

$$x_i^w = L \cdot x_i^{'} + x_{oi} \tag{2.3}$$

By applying converse process, we can extract and restore the original image.

2.4.2 Reversible watermarking by applying difference expansion

Other than replacing the LSBs in the original image to embed watermark in the first kind of watermarking algorithm, another way is to make bit-shifting to the values and release vacancy for watermark bits. Take an 8-bit value for example:

In this case, we get a vacancy at the 8th bit plane to embed a watermark bit.

In order to make this idea practical, the values need to be small so that the operation of bit-shifting will not introduce large change compared to the original. Base on this requirement, Tian [35] firstly proposed a pair-wise reversible integer transform used in watermarking to generate small values according to the original pixel values and shift

the bit plane of those small values to embed watermark. After this kind of scheme is proposed, it is widely discussed in many papers such as [32]-[42]. In order to introduce the general process of this method, we present the idea from Allattar [33] for example:

Applying a vector $u = (u_0, u_1, u_2, \cdots, u_N)$ from N pixels and make the integer transform to this vector which is demonstrated as follows:

$$\begin{aligned} v_0 &= \left\lfloor \frac{\sum_{i=0}^{N-1} u_i}{N} \right\rfloor && (2.4) \\ v_1 &= u_1 - u_0 && (2.5) \\ &\vdots && (2.6) \\ v_{N-1} &= u_{N-1} - u_0 && (2.7) \end{aligned}$$

where $\lfloor x \rfloor$ denotes the largest integer no larger than x and u_i denotes the values of N different pixels. In order to make use of the correlations of the near-by points, these N pixels are always picked as adjacent ones.

Similar to the pair-wise processing, The inverse transform which can regenerate the vector u from v is as follows:

$$\begin{aligned} u_0 &= v_0 - \left\lfloor \frac{\sum_{i=1}^{N-1} v_i}{N} \right\rfloor && (2.8) \\ u_1 &= v_1 + u_0 && (2.9) \\ &\vdots && (2.10) \\ u_{N-1} &= v_{N-1} + u_0 && (2.11) \end{aligned}$$

In this case, we can embed N-1 bits ($b = (b_1, b_2, \cdots, b_{N-1})$) into vector v by left-

shifting the difference values we get from the above equations (i.e. difference expansion).

$$v_0^{'} = \left\lfloor \frac{\sum_{i=0}^{N-1} u_i}{N} \right\rfloor \tag{2.12}$$

$$v_1^{'} = 2 \times v_1 + b_1 \tag{2.13}$$

$$\vdots \tag{2.14}$$

$$v_{N-1}^{'} = 2 \times v_{N-1} + b_{N-1} \tag{2.15}$$

After inverse integer transform is applied to the difference vector v' in which the watermark is embedded, we can get the watermarked pixels' values.

With the consideration of avoiding overflow and underflow, not all of the pairs or vectors can be chosen as expandable ones to embed watermark. A location map is necessary to tell the receiver which vectors are changed and this part of information is also appended in the watermark bits. In this case, the smaller size of the location map, the more space is left for the watermark bits. And obviously, the more elements in a vector, the smaller size of the location map is. However, the more elements in one vector, the larger value will be generated after integer transformation and the watermarked image will have severer artifacts. In this case, the number of elements in one vector should be chosen carefully so that good balance between capacity and invisibility can be achieved.

2.4.3 Reversible watermarking by exchanging the histogram bins

The last category of reversible watermark is to embed the information by exchange the histogram bins in the original image. To demonstrate the basic idea of this kind of scheme, we present Vleeschouwer et al.'s circular interpretation method [44][45].

First of all, the original image is divided into several un-overlapped blocks.

Then one watermark bit is embedded into each block by changing the histogram of the block. The process of embedding one bit is discussed as follows:

1. Assuming a block B, it is divided into two groups randomly, and the histogram H_a and H_b of each group are calculated respectively. For the reason that these two groups are divided randomly, the histogram of these two blocks should be similar and it is reasonable to assume that the peak bin of H_a and H_b are located at the same intensity value.

2. When we embed the watermark bit 1, all of the histogram bins are left-shifted except replacing the lowest bin to the location of the former highest bin; and when we embed the watermark bit 0, all of the histogram bins are right-shifted except replacing the highest bin to the location of the former lowest bin.

The embedding process can be shown in Fig. 2.4.

After repeating this process to all of the blocks, the watermark embedding is finished.

When we want to extract the watermark, we just need to reconstruct the two groups in each block and check the direction of the shifting of peak histogram to determine whether a bit 0 or bit 1 is embedded. After that, we can restore the original image by shifting the histogram back to its original place.

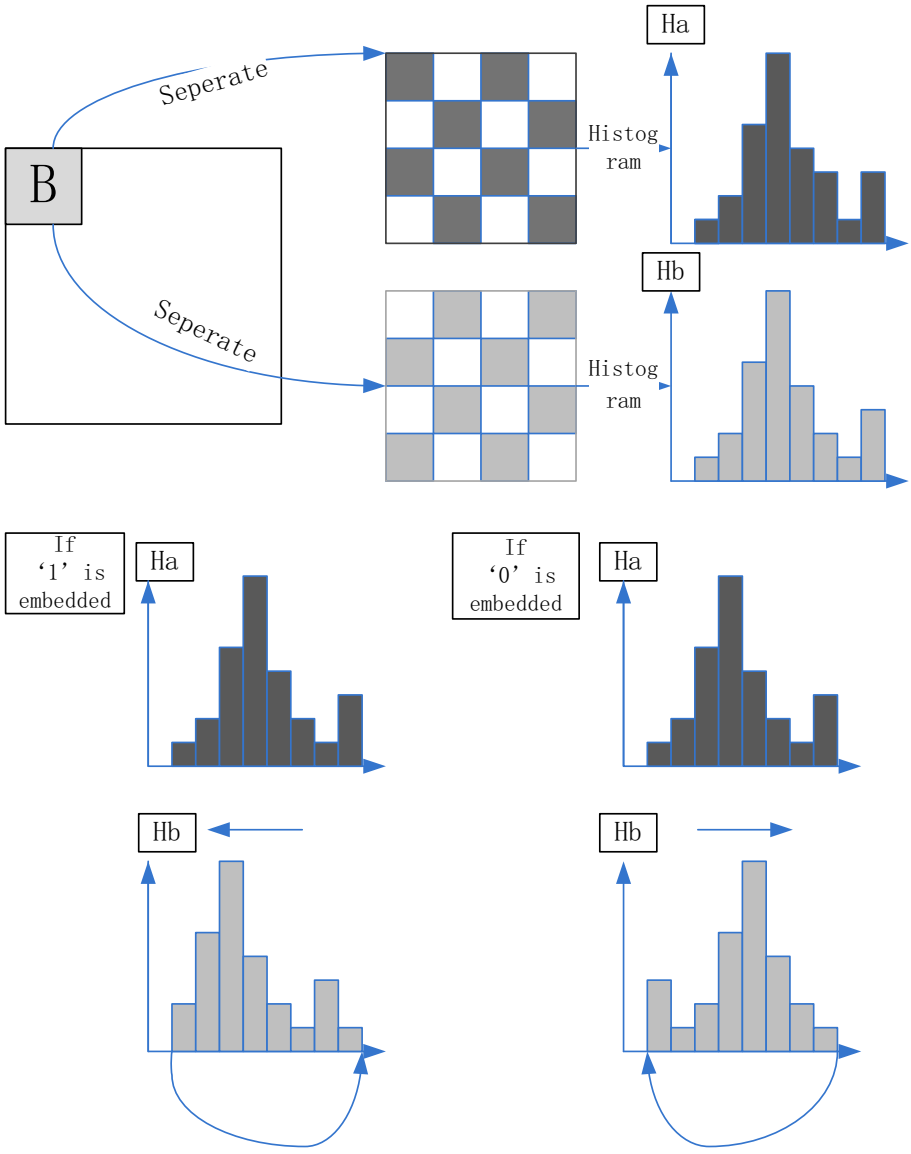


Figure 2.4: Watermarking by exchanging the histogram bins.

2.4.4 Comparison of reversible watermarking schemes

In the case of our application of watermarking, a very high capacity is required while the high visual quality is also desired. The characteristics of reversible watermarking seem to satisfy our requirement because the original image can be restored and we do

not have to concern the quality distortion after watermark embedding as much as the permanent distortion watermarking algorithms.

And meanwhile we need to make some comparison between these three reversible watermarking methods to determine which one is the most suitable scheme for our application.

For the first scheme which depending on the data compression, the capacity is highly depending on the compression ratio of the original image. However the host image in our expecting application is in JPEG format which is already compressed and there is little space for further lossless compression.

For the third scheme which applies histogram shifting, one watermark bit is embedded into one block of the original image. In this case, the capacity of this algorithm is much lower than the other two.

For the second scheme which applies the bit shifting and difference expansion method, its performance does not depend on the data compression as much as the first one, and N-1 watermark bits can be embedded in a N bits block which give us a larger capacity compared to the third kind of reversible watermarking.

In this case, we decide to choose the reversible watermarking algorithm based on bit shifting as our method to achieve the goal to embed the depth map into its corresponding 2D image in JPEG format.

2.4.5 The performance of the state of the art DE based watermarking algorithm

After Tian firstly proposed the difference expansion method on watermarking, lots of researches have been done based on it, and our purpose is to determine the probability

of using this scheme to hide depth map into one frame of its associated stereo images. In this case, we focus on the high capacity situation with a relatively low PSNR, for the reason that the original high quality image will be restored at the receiver side when the hidden depth map is extracted before reconstructing the 3D scene. From [32] to [42], they are all watermarking methods based on difference expansion, in order to get a general view of the capacity for this category of watermarking algorithm, we list some experimental results based on DE watermarking method in [32] to [42].

Table 2.1: Capacity of existing DE watermarking

Paper	*Lena*	*Baboon*	*Airplane*
A.M.Alattar[32]	2.17	0.44	NP
JunTian[35]	0.99	NP	NP
D.M.Thodi[36]	0.95	0.55	0.95
SunilLee[37]	1+	NP	NP
VasiliySachnev[38]	1+	0.61	1+
X.Wang[39]	0.9	0.5	NP
S.Weng[40]	1+	0.56	NP
YongjianHu[41]	0.98	0.51	0.95
C.C.Chang[42]	0.98	0.72	NP

In Table 2.1, NP (not presented) means there is no experimental results for that image in the particular paper.

We notice that for the image whose contend is intensive, the capacity is relatively small and the value is about 0.5-0.7 bpp while the capacity is rather large for the flat images with a value over 1 bpp.

We recall that the requirement of the capacity for embedding the depth map into its corresponding 2D image is less than 0.5 bpp from Section 2.3. It is hopefully that we can embed the depth map into its associated 2D image using bit shifting and difference expansion.

Chapter 3

The proposed scheme and its implementations

3.1 High capacity embedding using reversible watermarking with DE

To satisfy the requirement of the reduction of the transmission bandwidth and storage space for 3D vision media source, a straightforward approach is to simply compress and encode the stereo media pair separately and the storage requirement are nearly twice as much as the requirement of the single content [6]. An alternative approach utilizes the disparity estimation and disparity compensation schemes. The basic idea of this approach is to use the stereo media pair to generate a disparity map or a depth map which can be used to estimate the other stereo image with regarding to only one of stereo image pair. In this case, only the disparity map, one of the stereo image and sometimes the estimation error are encoded, transmitted and stored [7].

Furthermore, some researchers considered to utilize the watermarking methods to

hide the depth map into one of the stereo image for the purpose of enhancing the security of the depth information and reducing the transmission bandwidth and storage requirement.

As introduced in Section 2.1, all of the researchers used the reversible watermarking to hide the depth map into the stereo image in their proposals. The advantage of this kind of watermarking algorithm is not only the high capacity, which is essential to hide the whole depth map, but also the characteristic of retrieving the original image quality before watermarking, which can promise high image quality after heavy payload was embedded into the cover image [8][9][10][12][13][14].

In this case, we firstly implemented a classical reversible watermarking algorithm using difference expansion (DE) in spacial domain to investigate its capacity and anti-compression characteristics.

As introduced in [33], an improved difference expansion method for reversible watermarking is implemented as shown in Fig. 3.1:

The watermark embedding is processed as follows:

The cover image is divided into 2×2 blocks and each block is the basic structure in the embedding scheme. Assigning a vector $u = (u_0, u_1, u_2, u_3)$ to 4 pixels values in each 2×2 block and make the integer transform to this vector which is demonstrated as follows:

$$v_0 = \left\lfloor \frac{\sum_{i=0}^{3} u_i}{4} \right\rfloor \tag{3.1}$$

$$v_1 = u_1 - u_0 \tag{3.2}$$

$$v_2 = u_2 - u_0 \tag{3.3}$$

$$v_3 = u_3 - u_0 \tag{3.4}$$

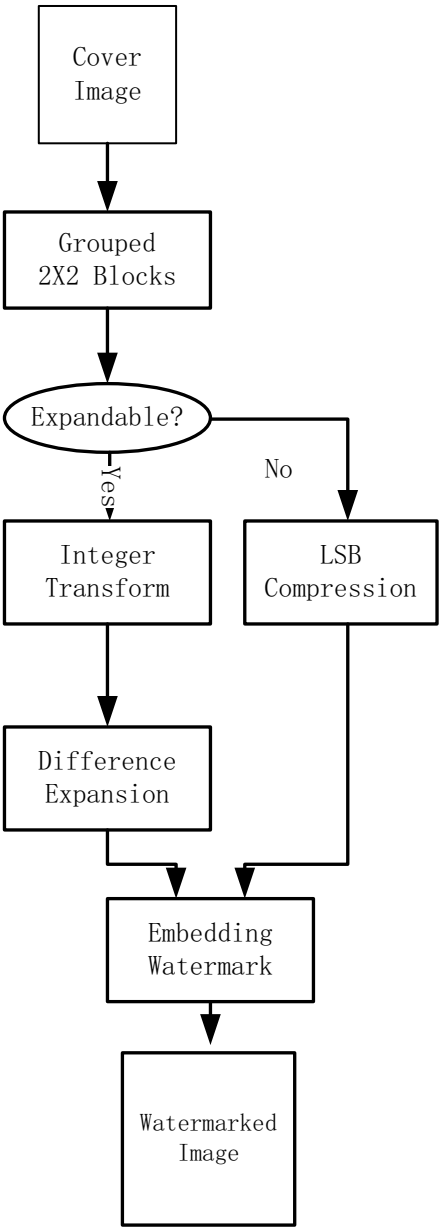


Figure 3.1: Watermarking by difference expansion and LSB compression.

where $\lfloor x \rfloor$ denotes the largest integer no larger than x and u_i denotes the values of 4 different pixels. In order to make use of the correlations of the near-by points, these 4 pixels are always picked as adjacent ones.

In order to keep the visual distortion of the watermarking small, a threshold K is set for the difference values (v_1, v_2, v_3). Only the blocks with the difference values smaller than K is allowed to embed the watermark bits. For the expandable blocks, we can embed 3 bits ($b = (b_1, b_2, b_3)$) into vector v by left-shifting the difference values we get from Equ. (3.1–3.4) (i.e. difference expansion).

$$
\begin{aligned}
v_0^{'} &= \left\lfloor \frac{\sum_{i=0}^{3} u_i}{3} \right\rfloor && (3.5)\\
v_1^{'} &= 2 \times v_1 + b_1 && (3.6)\\
v_2^{'} &= 2 \times v_2 + b_2 && (3.7)\\
v_3^{'} &= 2 \times v_3 + b_3 && (3.8)\\
& && (3.9)
\end{aligned}
$$

Similar to the pair-wise processing, The inverse transform, which can regenerate the vector u from v, is as follows:

$$
\begin{aligned}
u_0 &= v_0 - \left\lfloor \frac{\sum_{i=1}^{3} v_i}{4} \right\rfloor && (3.10)\\
u_1 &= v_1 + u_0 && (3.11)\\
u_2 &= v_2 + u_0 && (3.12)\\
u_3 &= v_3 + u_0 && (3.13)
\end{aligned}
$$

After inverse integer transform is applied to the difference vector v' in which the watermark is embedded, we can get the watermarked pixels' values.

For the block whose difference values are greater than the threshold K, we applied

lossless compression to the LSB of the pixel values' in that block. And then, the watermark bit plus the compressed LSB is embedded together in the original location of the LSB plane.

The watermarking extraction is processed as follows:

At the receiver side, the blocks with DE watermark embedding and LSB watermark embedding are identified by the labels which are also embedded in the cover image.

For the block with LSB watermark embedding, The bits in the LSB of the pixel values in the block is extracted. The extracted bit sequence are combined with two parts: the embedded watermark bits and the compressed original bits in the LSB. After the decompression of the original LSB, the original blocks before the watermark embedding can be recovered.

For the block with DE watermark embedding, the integer transform is applied to the block. In the transformed domain of the block, the watermark bits are in the LSB of the transformed values. After right-shifting the transformed values and inverse transform, the original block can be recovered.

After applying the watermark extraction and original block recovering to all of the blocks, the original cover image can be obtained and the watermark bits can be extracted completely.

Because of the property of reversible watermarking, the embedding process can be done iteratively. It means that the watermarked image can be watermarked with the same scheme again and again until the visual quality is not accessible or the difference of the adjacent pixel value are too large for embedding.

We chose 5 images with the size of 512×512 to embed the watermark bits using the reversible watermarking based on spacial difference expansion. The implementation results are shown in Fig. 3.2.

Figure 3.2: Watermarking using DE in spacial domain.

In Fig. 3.2, the watermarked images hiding different watermark bits are compared. In the left column, the watermark payload is 0.0625 bpp; in the middle column, the watermark payload is 0.35 bpp; in the right column, the watermark payload is 0.9 bpp.

The PSNR value of the watermarked images are shown in Fig.3.3.

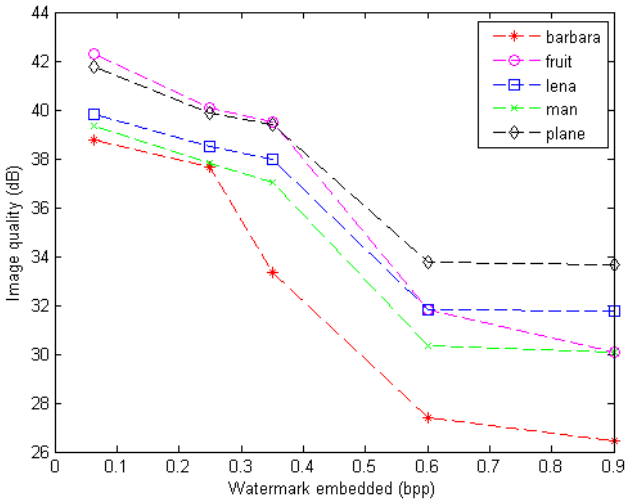


Figure 3.3: PSNR versus embedding payload.

It can be observed that the capacity of this kind of watermarking algorithm is very high. In our implementation, all of the five test images can obtain an capacity beyond 0.9 bpp. Although the PSNR is low with high embedding payload, the original cover image can be retrieved after the watermark extraction and the high visual quality of the original image can be preserved.

However, the watermark extraction can not be successfully implemented if we compress the watermarked image using JPEG standard. This problem exists in all of the existing approaches for depth map hiding using reversible watermarking. In order to overcome this problem and make the embedding process compatible with JPEG compression, we applied the DE algorithm into the quantized DCT domain of the image.

3.2 DE watermarking in the quantized DCT domain

In the previous section, we found that the existing approaches, which hide the depth map in the spatial domain, is not compatible with the JPEG compression. With the concern that the JPEG format image is widely used nowadays, we attempt to find a solution to hide the depth map into a JPEG format image.

In order to hide the depth map into a JPEG image using the high capacity reversible watermarking which is fragile to distortion. we proposed to hide the depth map into the quantized DCT domain of the image. The reason for this proposal is that the quantization of the DCT coefficients in JPEG compression will remove most of the slight changes introduced by watermarking. In this case, the watermark information will not be removed if we embed the watermark after the quantization process which is the only lossy procedure in JPEG compression.

Furthermore, we only embed the watermark in the blocks whose DCT values are not zero. This will guarantee the compression efficiency of the Huffman coding in the later processing.

The embedding process is introduced in Fig.3.4.

In our proposed watermarking scheme, the watermark is embedded by using the following steps:

1. Transform 2D un-compressed image to DCT coefficients.

2. Quantize each of the DCT coefficients using a predefined quantization matrix.

3. Divide all of the quantized DCT coefficients into 2×2 vectors and pick out all the vectors (u) whose values are not zero.

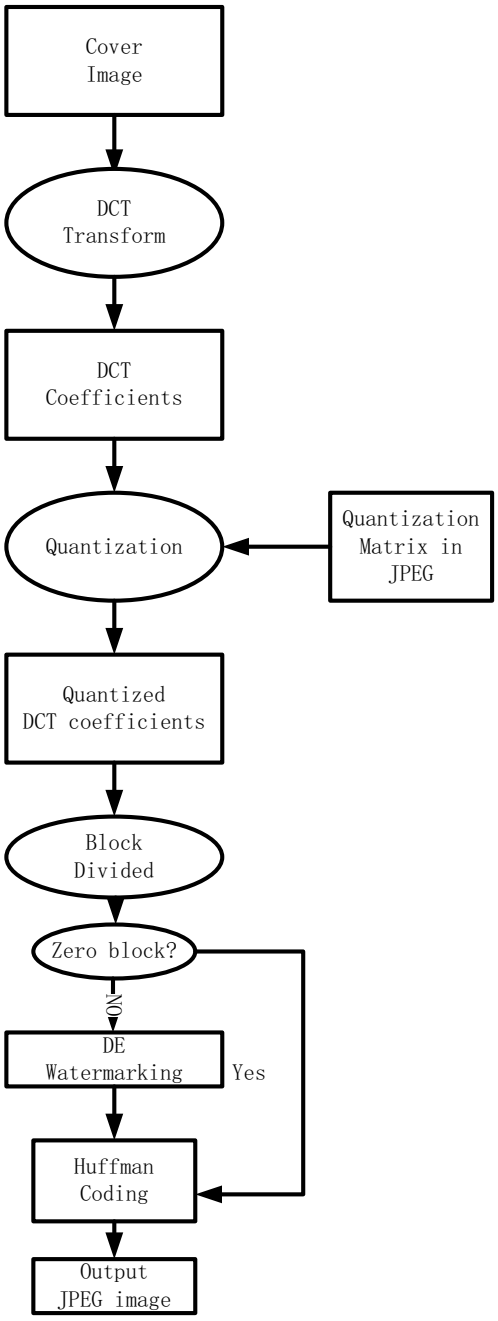


Figure 3.4: Embedding process of DE watermarking in quantized DCT domain.

4. Apply generalized integer transform to the vectors (u) picked out in step 3 to generate the new vectors (v)

5. Embed the watermark by left-shifting the elements in vector (v) and add the watermark bit in the LSB of the shifted values.

6. Apply in these integer invert transform and encode the watermarked DCT coefficients by Huffman coding.

The watermark is extracted by using the following steps:

1. Apply Huffman decoding to the received signal and get the watermarked DCT coefficients.

2. Pick out all the vectors containing non-zeros values.

3. Apply generalized integer transform to those vectors.

4. Extract the watermark by taking the LSBs of $v_1, v_2, \cdots v_{N-1}$ in vector v . Restore the original transformed integer by right-shifting the bit plane of $v_1, v_2, \cdots v_{N-1}$.

5. Apply inverse integer transform to the restored vector v to obtain the original DCT coefficients.

We implement this watermarking scheme for five images with different watermark bits embedded. It can be observed in Fig. 3.5 that the visual quality of the watermarked images is very low. However, the original image can be restored after watermark detection. In this case, we can still obtain high visual quality image after a heavy load of watermark bit embedded.

In Fig. 3.5, the watermarked images hiding different watermark bits are compared. In the left column, the watermark payload is 0.0095 bpp with one level embedding; in the middle column, the watermark payload is 0.2384 bpp with two levels embedding; in the right column, the watermark payload is 0.4673 with three level embedding bpp.

Another problem of the current scheme is that the size of the watermarked image is increased significantly with regard to the embedding watermark size.

Table 3.1: DE embedding in quantized DCT domain of image Barbara

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Results	Size increase(bpp)	0.0224	0.5109	1.0436
	Increase ratio	2.3548	2.1428	2.2333

Table 3.2: DE embedding in quantized DCT domain of image Fruit

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Results	Size increase(bpp)	0.0188	0.4755	0.9708
	Increase ratio	1.9728	1.9943	2.0775

Table 3.3: DE embedding in quantized DCT domain of image Lena

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Results	Size increase(bpp)	0.0227	0.5608	1.1566
	Increase ratio	2.3880	2.3523	2.4751

Table 3.4: DE embedding in quantized DCT domain of image Man

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Results	Size increase(bpp)	0.0187	0.4902	1.0217
	Increase ratio	1.9584	2.0562	2.1864

From Table 3.1-3.5, it can be observed that the ratio between the size of watermarked image and the embedded watermark varies from 2 to 3. In this case, our current

Table 3.5: DE embedding in quantized DCT domain of image Plane

		Embedding Level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Results	Size increase(bpp)	0.0265	0.6666	1.2892
	Increase ratio	2.7840	2.7961	2.7588

watermarking scheme introduces additional size increasement to the stereo image other than the depth information which is embedded.

In this case, we customize the Huffman coding scheme to make it more efficient to code our watermarked DCT coefficients.

3.3 Improvement: Huffman table customization

In information theory, Huffman coding is an entropy encoding method which are used for lossless data compression. A variable-length code table is used to encode the source symbol. The encoding table is derived depending on the estimated probability of the occurrence of each possible symbol. If the frequency of the signal occurrence is known, a Huffman table can be generated so that the most common signals are encoded with the shorter strings of bits while the less common signals are encoded with longer strings of bits.

For natural image, most of the energy is in the low frequency components. In this case, most of the quantized DCT coefficients are very small value. With regarding to this characteristic, the default Huffman table in JPEG assigned shorter length codes for small values and longer strings of bits to large values.

However, the distribution of the quantized DCT coefficients are changed after watermarking. In this case, the signals, witch occurs most frequently, are not always the

small values. As a result, the default Huffman table is not optimized for the watermarked image. In our improvement, the Huffman table is customized according to the distribution of the quantized DCT coefficients.

After the Huffman table customization, we get the same watermarked image with smaller size compared to the previous approach.

Table 3.6: DE embedding in quantized DCT domain of image Barbara (with Huffman table customization)

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Size increase	With customization (bpp)	2.3548	2.0852	2.2295
	Without customization (bpp)	2.3548	2.1428	2.2333

Table 3.7: DE embedding in quantized DCT domain of image Fruit (with Huffman table customization)

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Size increase	With customization (bpp)	1.9728	1.9822	2.0148
	Without customization (bpp)	1.9728	1.9943	2.0775

Table 3.8: DE embedding in quantized DCT domain of image Lena (with Huffman table customization)

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Size increase	With customization (bpp)	2.3880	2.2948	2.3178
	Without customization (bpp)	2.3880	2.3523	2.4751

It can be found that the size increase of the cover image is still over twice as much as the amount of watermark bits embedded. In order to obtain a lower ratio between the cover size increase and the watermark bit embedded, the stochastic characteristic of the quantized DCT coefficients are analyzed in the following section.

Table 3.9: DE embedding in quantized DCT domain of image Man (with Huffman table customization)

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Size increase	With customization (bpp)	1.9584	1.9789	2.0613
	Without customization (bpp)	1.9584	2.0562	2.1864

Table 3.10: DE embedding in quantized DCT domain of image Plane (with Huffman table customization)

		Embedding level		
		1	2	3
Input	Embedded watermark(bpp)	0.0095	0.2384	0.4673
Size increase	With customization (bpp)	2.7840	1.7844	2.5154
	Without customization (bpp)	2.7840	2.7961	2.7588

3.4 Improvement: DCT distribution preservation

The DCT coefficients of images obey a Laplacian-shape-like distribution [46]. This characteristic benefits the compression gain of Huffman coding for the reason that most of the energy distributed in a few ranges of low frequencies. In this case, only a few number of ranges of DCT values are needed to be encoded. And ideally, these ranges can be encoded by short strings of bits. However, the difference expansion changes the coefficient distribution and make the DCT coefficient distribution flat. In this case, there are more number of ranges of DCT coefficients to be encoded. Therefore, some of them have to be encoded by longer strings of bits compared to the natural image. The distribution change makes the JPEG compression less efficient. In this case, we further change our watermarking scheme to preserve the Laplacian-shape-like distribution of the DCT coefficients after the watermark is embedded into the cover image.

In Fig. 3.8, the upper row of figures are the distributions of the original image 'Barbara', the lower row of figures are the distribution of the watermarked images with

different embedding levels.

The reason for the distribution change lies on the integer transform of the quantized DCT coefficients. And besides, the motivation of applying integer transform is to generate small values which can be bit-shifted with smaller distortion. However, the DCT coefficients are already very small after quantization and it is not necessary to apply integer transform to these quantized coefficients. In this case, we change our watermarking scheme from difference expansion to directly bit-shifting the quantized DCT coefficient. As a result, the coefficient distribution structure is preserved after watermarking. Although some researchers have already worked on the bit-shift watermarking in DCT domain [47], these works did not consider the quantization of DCT coefficients that is used in JPEG compression.

Since the watermarking scheme is reversible, we can apply recursive watermark embedding. For one level embedding, the embedding equation is as follows:

$$Q_i^{'} = 2 \times Q_i + b_i \tag{3.14}$$

where Q_i is the quantized DCT coefficient, b_i is the watermark bit.

In order to keep the coefficient distribution unchanged, we have to change all of the non-zero coefficients in one-level embedding. If the watermark bits are not enough to change all of the coefficients, the unchanged coefficients are left-shifted directly.

After applying the bit-shifting to the quantized DCT coefficients, the Laplacian-shape-like distribution of the quantized DCT coefficients is successfully preserved after the watermark embedding. The different DCT coefficient distribution is demonstrated in Fig. 3.9.

In Fig. 3.9, the horizontal axis denotes the different intervals in which the AC

coefficients lay in, and the vertical axis denotes the amount of the AC coefficients in that interval. For each group of figures, the upper one is the coefficients distribution after watermarking using DE and the lower one is the coefficients distribution after watermarking using bit-shifting. It is obviously that the bit shifting procedure preserves the coefficients distribution and the only change to the distribution is the right-shifting which can be fixed by Huffman table customization. As the result, the image size increase introduced by watermarking can be significantly reduced because the coefficients still distribute in a few number of values ranges which can benefit the Huffman compression efficiency.

In Fig. 3.10, watermark bits are embedded into four test images with different payload. Since the reversible watermark embedding process can be repeated iteratively for a single image, we applied three different embedding levels with different watermarking capacity. For the left column of the images, each of them were embedded by one level embedding; for the middle column of the images, each of them were embedded by two level embedding; and for the right column of the images, each of them were embedded by three level embedding.

The detailed information about the watermarking in four test images are shown in Table. 3.11-Table. 3.14.

Table 3.11: BS watermarking embedding in image Barbara

		Embedding level					
		1		2		3	
Input	Embedded watermark(bpp)	0.0625	0.1500	0.2300	0.3000	0.3500	0.4500
Results	PSNR(dB)	20.30	20.30	14.39	14.31	11.30	11.25
	Size increase(bpp)	0.1500	0.1500	0.3000	0.3000	0.4500	0.4500

From Table. 3.11-Table. 3.14, it can be observed that the size increase in a single embedding level is constant and the capacity in each embedding level is almost the

Table 3.12: BS watermarking embedding in image Fruit

		Embedding level					
		1		2		3	
Input	Embedded watermark(bpp)	0.0380	0.1570	0.2355	0.3140	0.3925	0.4710
Results	PSNR(dB)	22.73	22.14	16.02	16.02	12.08	12.03
	Size increase(bpp)	0.1572	0.1572	0.3143	0.3143	0.4715	0.4715

Table 3.13: BS watermarking embedding in image Lena

		Embedding level					
		1		2		3	
Input	Embedded watermark(bpp)	0.0380	0.0960	0.1440	0.1930	0.2400	0.2880
Results	PSNR(dB)	23.83	23.52	16.90	16.80	13.08	13.06
	Size increase(bpp)	0.0969	0.0969	0.1937	0.1937	0.2906	0.2906

Table 3.14: BS watermarking embedding in image Plane

		Embedding level					
		1		2		3	
Input	Embedded watermark(bpp)	0.0380	0.0900	0.1350	0.1800	0.2250	0.2700
Results	PSNR(dB)	23.13	22.90	17.03	16.21	13.76	13.64
	Size increase(bpp)	0.0907	0.0907	0.1814	0.1814	0.2721	0.2721

same as the size increase of the cover image.

This result is much better than the previous result from embedding the watermark by using DE in quantized DCT domain.

After the above improvements and modifications to our proposal, our watermarking scheme can be applied to JPEG format image. In this case, the size of the watermarked image in our scheme is significantly decreased compared to existing approaches which embedded the depth map into the un-compressed spatial domain. And the size of the watermarked image is almost equal to the sum of the size of the original JPEG image plus the size of the watermark embedded. As a result, we successfully embedded a large amount of informatino into a JPEG compressed image without introducing noticeable extra size and the cover image can be completely restored after the watermark is ex-

tracted at the receiver side. And besides, the watermarked image can be re-compressed by the process of watermark extraction, original image restoration, watermark embedding and re-compression.

3.5 Apply reversible watermarking scheme to MPEG-2 compressed video

After we successfully embedded a large amount of watermark into the non-zero components of the quantized DCT domain of the JPEG format image, we try to apply our watermarking scheme to MPEG-2 compression videos.

The compression scheme is quite similar between JPEG compression and MPEG-2 compression, especially the procedures of DCT transformation and quantization, on which our watermarking method relies, are used in both of these two compression methods. In this case, a compressed depth video can be hidden into its corresponding stereo video and the watermarked cover video can also be restored after the depth information is extracted.

In a MPEG-2 compression system, the compression procedure is based on motion prediction (motion estimation at the encoder and motion compensation at the decoder) and two dimensional discrete cosine transform, DCT coefficients quantization, and Huffman coding.

There are three kinds of frames to be encoded in the MPEG-2 scheme:

1. I frame: it is intra coded frame. All macroblocks in this kind of frame is coded without prediction. It is the reference frame for other two frames. The compression ratio of this frame is the lowest.

2. P frame it is predicted frame. The macroblocks can be encoded with forward prediction using the previous I/P frames as reference. It also can be encoded by intra coded if the variance of a block is under certain threshold.

3. B frame: it is bi-directionally predicted frame. The macroblocks can be coded with forward prediction from previous frames or can be coded with backward prediction from future frame. This kind of frame has the highest compression ratio.

Since both of the internal encoded macroblock data and the predicted error should be transformed into DCT domain. And then the DCT coefficient quantization is utilized to remove the spacial redundancy of the encoded signal. In this case, we can still embed our watermark into the MPEG-2 compressed video by shifting the quantized DCT value of the inter coded blocks and the prediction error blocks.

In Fig. 3.15, our proposed video watermarking scheme is based on a contentional MEPG-2 encoding system. The watermarking process is inserted into the MPEG-2 encoding scheme, which is applied after the quantization of the DCT coefficients of the inter coded blocks or the prediction errors of the prediction blocks. After embedding the watermark with bit shifting, the watermarked coefficients are encoded with customized Huffman table.

For the watermark extraction, the watermark bits are extracted after the Huffman decoding. The extraction is very easy to implement as shown in Fig. 3.16.

Furthermore, the bit-rate control system can be utilized to further decrease the size of the watermarked image. After the bit-rate control system is utilized, the quantization parameter will change with regarding to the size of current encoded signals. If the bit number used to encode the signal is greater than a pre-defined threshold, the

quantization parameter for the further blocks will increase, so that the compression ratio will increase for the next blocks to be encoded. In this case, the size increase from watermarking will be balanced. The drawback of the bit rate control system is that the the size decrease is obtained by sacrificing the image quality which is introduced by a heavier compression ratio. And this kind of quality degradation can not be restored because it is introduced by lossy compression other than reversible watermarking.

Figure 3.5: Watermarking using DE in quantized DCT domain.

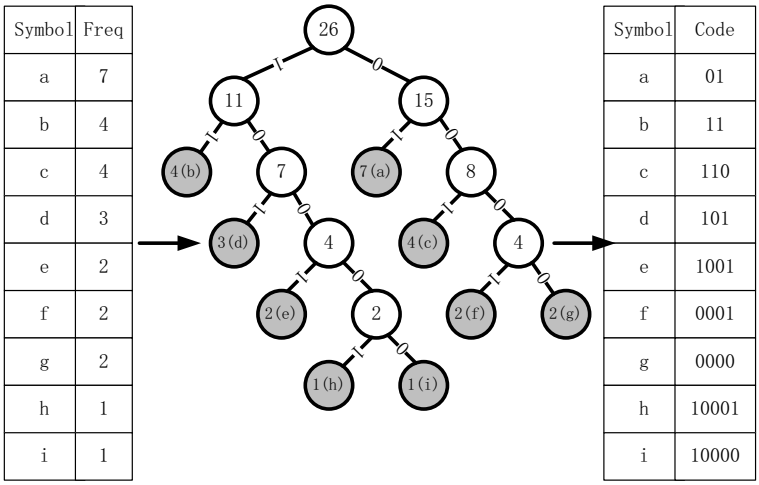


Figure 3.6: Huffman coding.

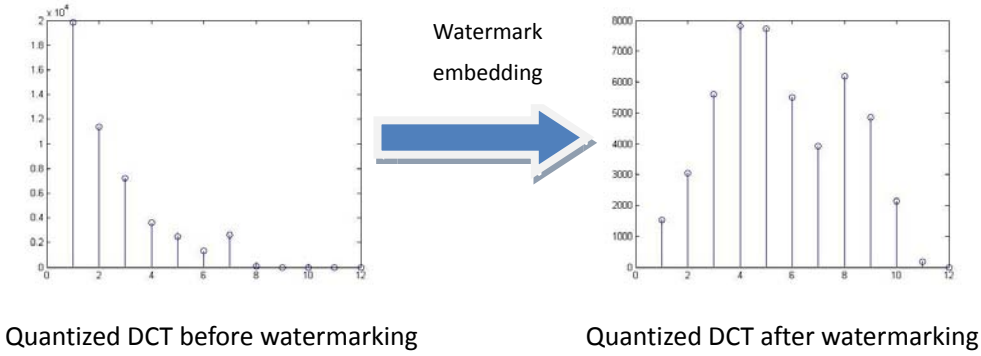


Figure 3.7: The distortion of DCT distribution.

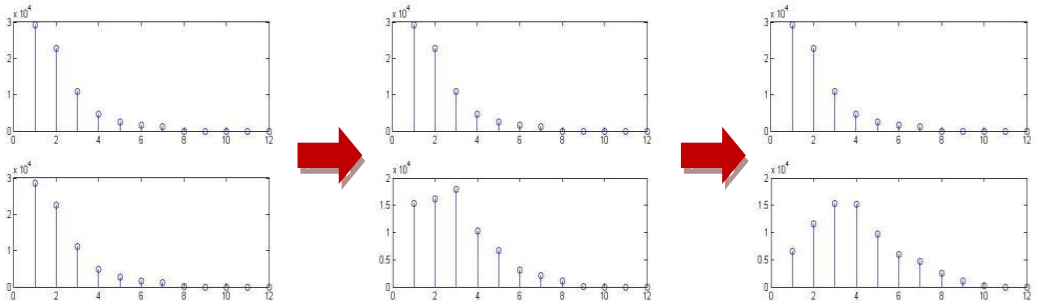
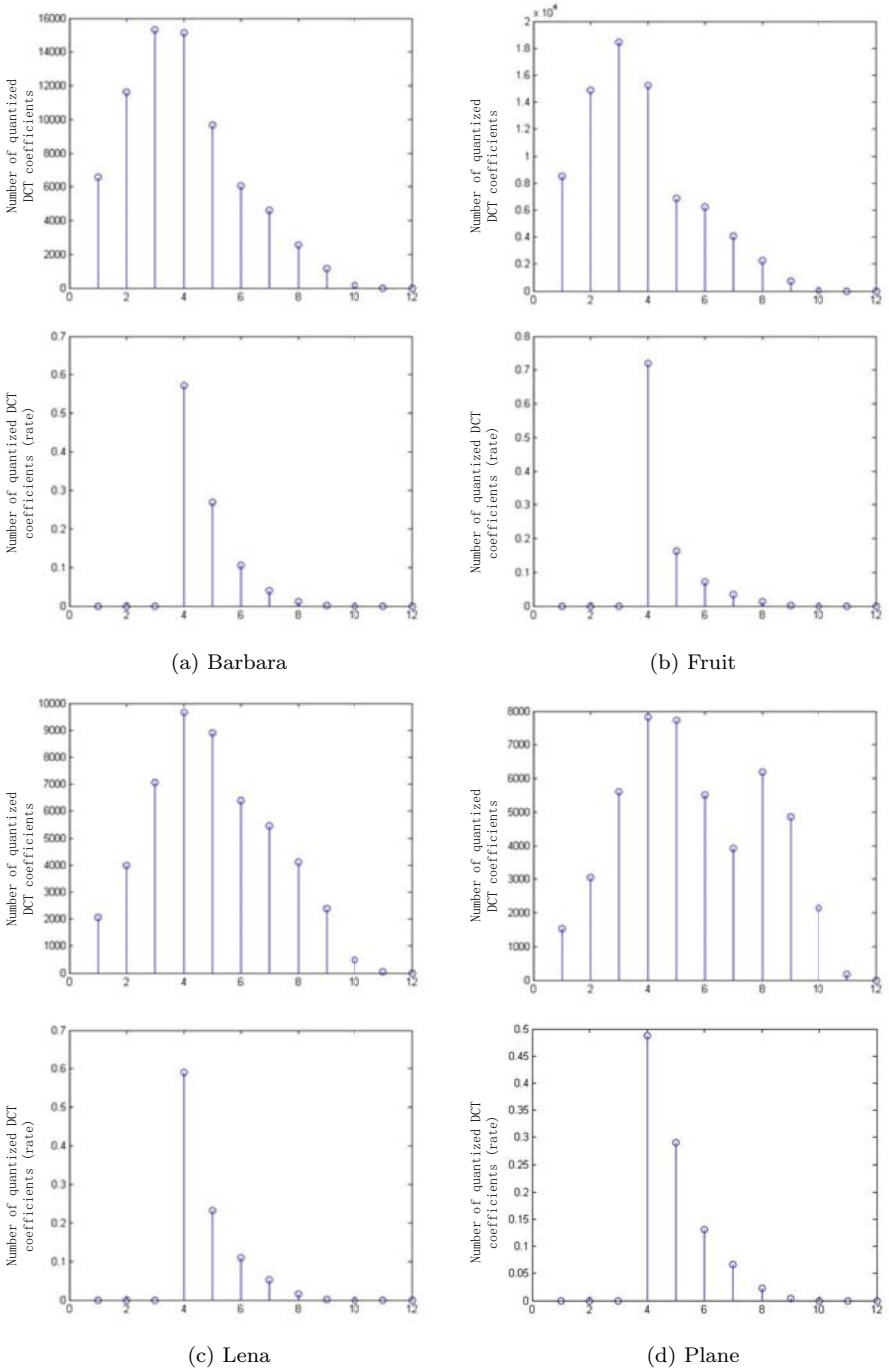


Figure 3.8: Coefficients distribution change after DE watermarking.



(a) Barbara

(b) Fruit

(c) Lena

(d) Plane

Figure 3.9: Comparison of the distribution of the DCT coefficients.

Figure 3.10: Watermarked images using Huffman table customization and bit-shift in quantized DCT domian.

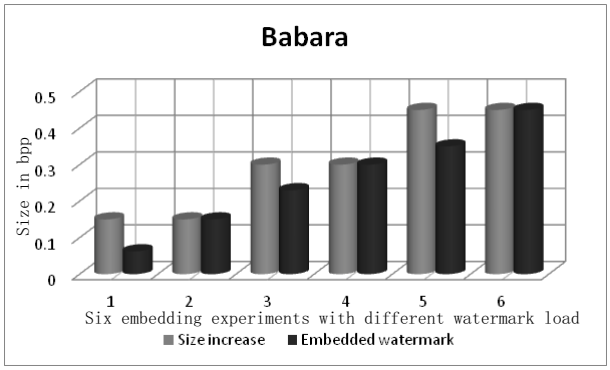


Figure 3.11: Barbara: the relationship between the size increase and watermark payload.

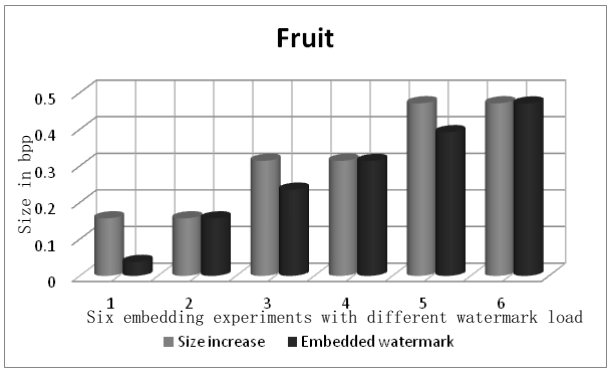


Figure 3.12: Fruit: the relationship between the size increase and watermark payload.

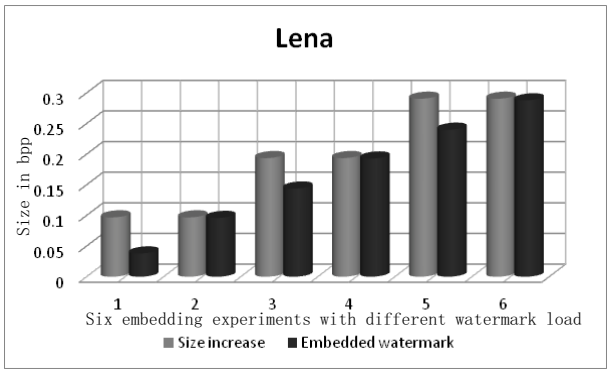


Figure 3.13: Lena: the relationship between the size increase and watermark payload.

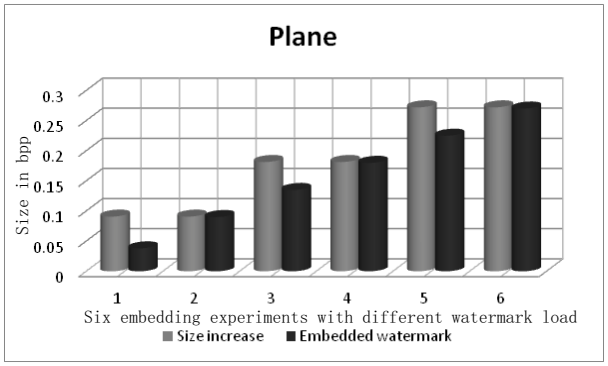


Figure 3.14: Plane: the relationship between the size increase and watermark payload.

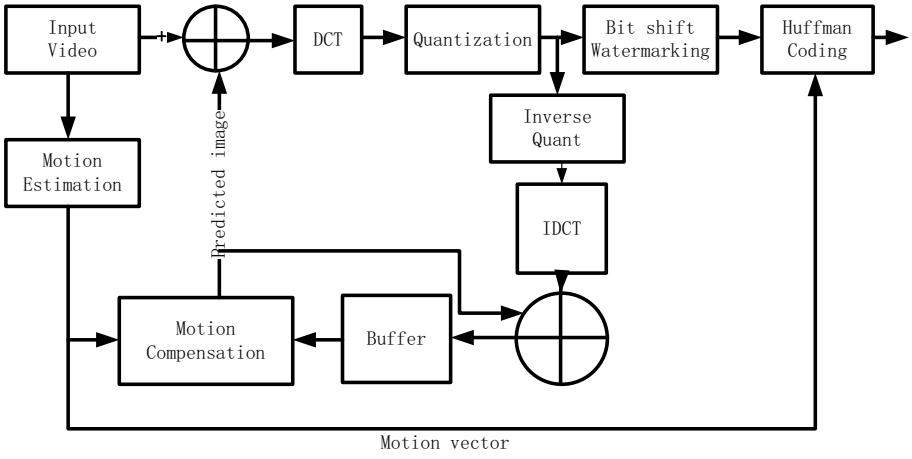


Figure 3.15: MPEG-2 encoder system.

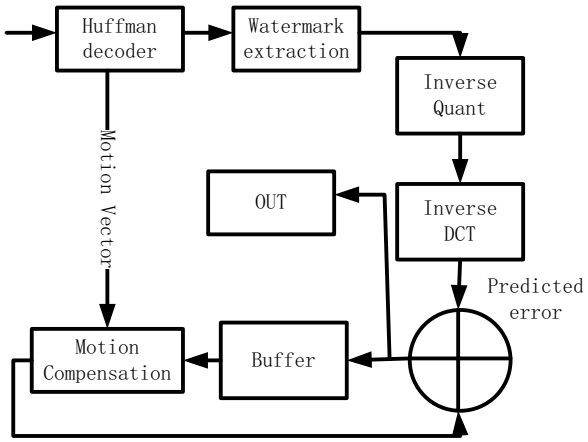


Figure 3.16: MPEG-2 decoder system.

Chapter 4

Experimental results

In this chapter, we illustrate the performance of the proposed scheme applied to hide the depth map in JPEG compressed color images and hide depth video into MPEG-2 compressed color video.

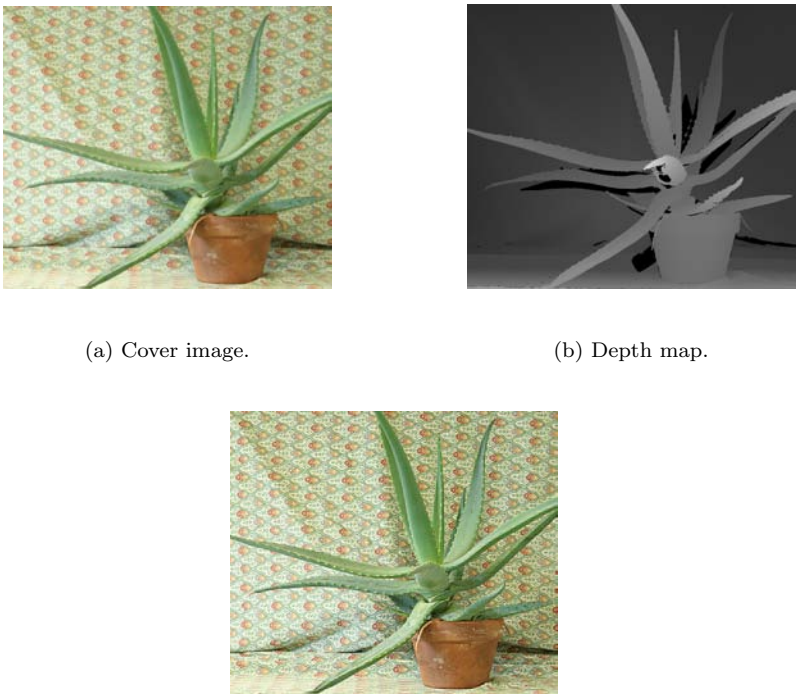
4.1 JPEG compressed images

We get a series of test data from [11]. In order to test the high capacity characteristic of our scheme, the depth map to be embedded is losslessly compressed in PNG format. The depth map is divided equally into three parts and each part is embedded into one channel of the stereo image corresponding to the depth map.

In Fig. 4.1, the depth map of size 73.5 KB is embedded into the cover image with one level embedding.

In Fig. 4.2, the depth map of size 83.2 KB is embedded into the cover image with two level embedding.

In Fig. 4.3, the depth map of size 54.8 KB is embedded into the cover image with



(a) Cover image.

(b) Depth map.

(c) Watermarked image.

Figure 4.1: Depth map hiding for image Aloe.

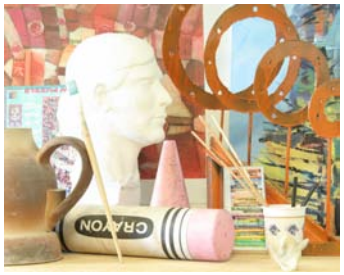
two level embedding.

In Fig. 4.4, the depth map of size 73.1 KB is embedded into the cover image with two level embedding.

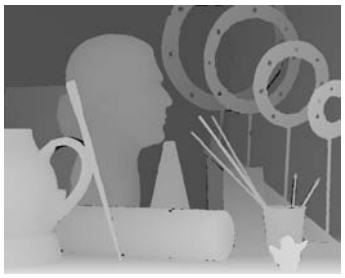
In Fig. 4.5, the depth map of size 65.9 KB is embedded into the cover image with three level embedding.

The detailed embedding information can be found in Table 4.1.

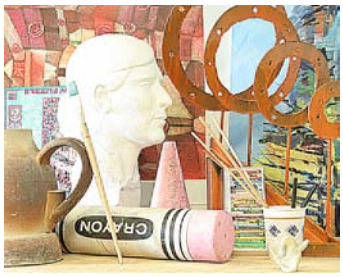
Table 4.1 shows the image size changes after watermarking. In this table, "Level" means how many embedding levels have been used. "Capacity/Level" means the capacity of each embedding level. Note that the product of "Capacity/Level" and "Level" which is the possible maximum load is almost the same as the size increase of the JPEG



(a) Cover image.



(b) depth map.



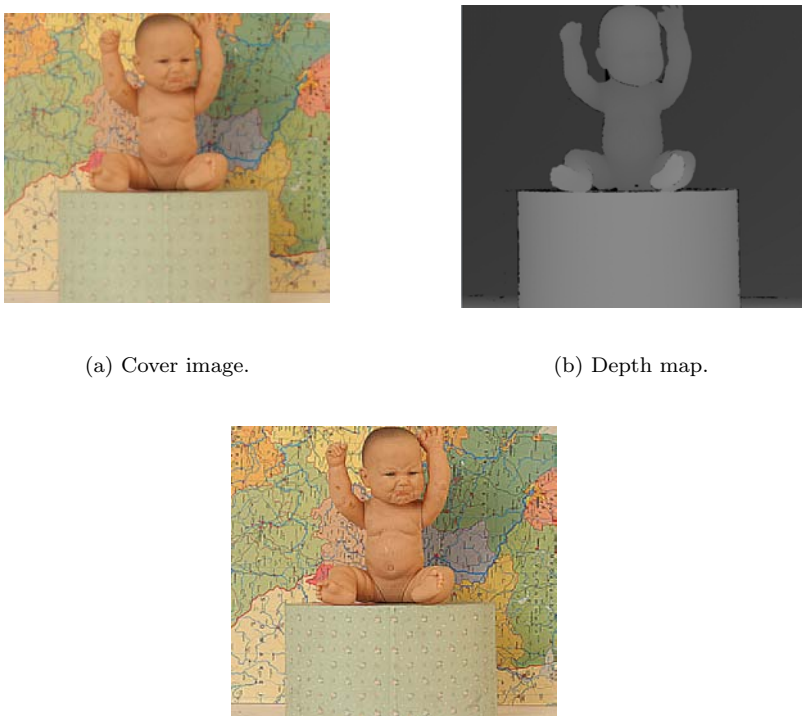
(c) Watermarked image.

Figure 4.2: Depth map hiding for image Art.

Table 4.1: Increase of image size after watermarking

	Aloe (1280×1104)		Art (1384×1104)		Baby (1240×1104)		Bowling (1328×1104)		Midd (1392×1104)	
	ori	wm	ori	wm	ori	wm	ori	wm	ori	wm
DC (Mbit)	0.376	0.376	0.376	0.376	0.297	0.297	0.295	0.295	0.263	0.263
AC (Mbit)	3.75	4.45	2.20	2.97	1.91	2.59	1.77	2.40	1.57	2.32
Size Increase (Mbit)	0.7		0.77		0.68		0.63		0.75	
Size Increase (bpp)	0.49		0.51		0.49		0.43		0.49	
Capacity/level (bpp)	0.49		0.25		0.25		0.22		0.16	
Level	1		2		2		2		3	
Watermark Load (bpp)	0.43 (73.5KB)		0.44 (83.2KB)		0.33 (54.8KB)		0.41 (73.1KB)		0.35 (65.9KB)	

image after watermarking. However, the real load (size of depth map) can not be always the maximum load for that embedding level and this makes the size of embedding information smaller than the capacity a little bit. In this case, the size of load is smaller than the size increase a little bit unless the payload is very close to the capacity for that embedding level.



(a) Cover image.

(b) Depth map.

(c) Watermarked image.

Figure 4.3: Depth map hiding for image Baby.

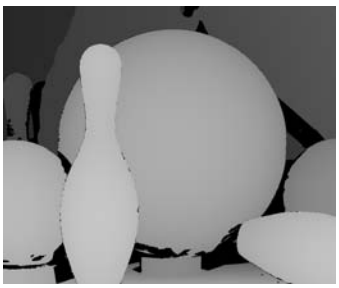
We also have done experiments on other test images, and the similar results have been achieved. Test results show that the scheme can meet the requirements of both high capacity and reversibility.

4.2 MPEG-2 compressed videos

We tested the stereo video sequence with the depth map information on [48]. After applying the same watermarking embedding scheme to the MPEG-2 video sequence, the compressed depth sequence was embedded into the quantized DCT coefficients.

As discussed in Section 3.5, the size of watermarked video will not increase as much

(a) Cover image.



(b) Depth map.

(c) Watermarked image.

Figure 4.4: Depth map hiding for image Bowling.

as that of the JPEG image because of the bit-rate control system in the MPEG-2.

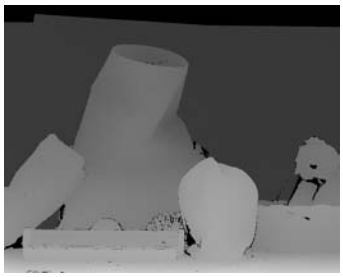
In our experiment, the depth map is compressed firstly with MPEG-4 algorithm, which can reach a compression ratio as high as 100 times. And meanwhile, the host video is compressed and watermarked in MPEG-2 scheme. The compression ratio of the MPEG-2 standards is around 20 times.

The watermarked video is shown in Fig. 4.7 to Fig. 4.9.

In Fig. 4.6 to Fig. 4.9, the compressed depth video is embedded into the host video in MPEG-2 format. Although the visual quality of the watermarked video is very low, the original quality can be restored after the watermark is extracted and a high quality video still can be obtained.

The detailed embedding information is demonstrated in Table 4.2.

(a) Cover image.



(b) Depth map.

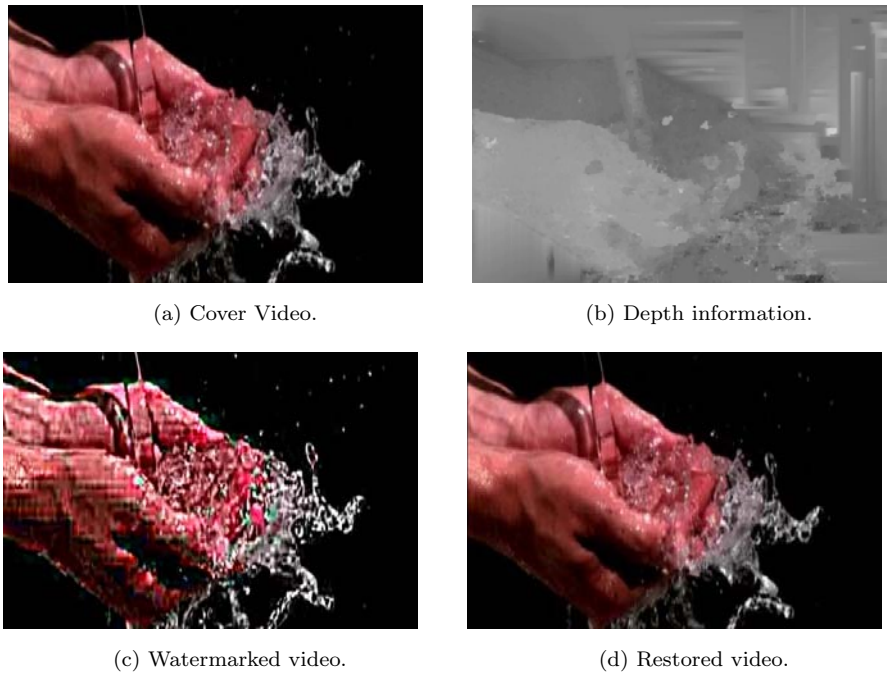
(c) Watermarked image.

Figure 4.5: Depth map hiding for image Mid.

Table 4.2: Watermarking of MEPG-2 video

	Horse	Hand	Car	Flower
Original Cover Size (KB)	590	587	587	588
Depth Video Size(KB)	73	82	90	62
Watermarked Video Size(KB)	590	589	588	588

We also have done experiments on other test videos, and the similar results have been achieved. Test results show that the scheme can meet the requirements of both high capacity and reversibility.



(a) Cover Video. (b) Depth information.

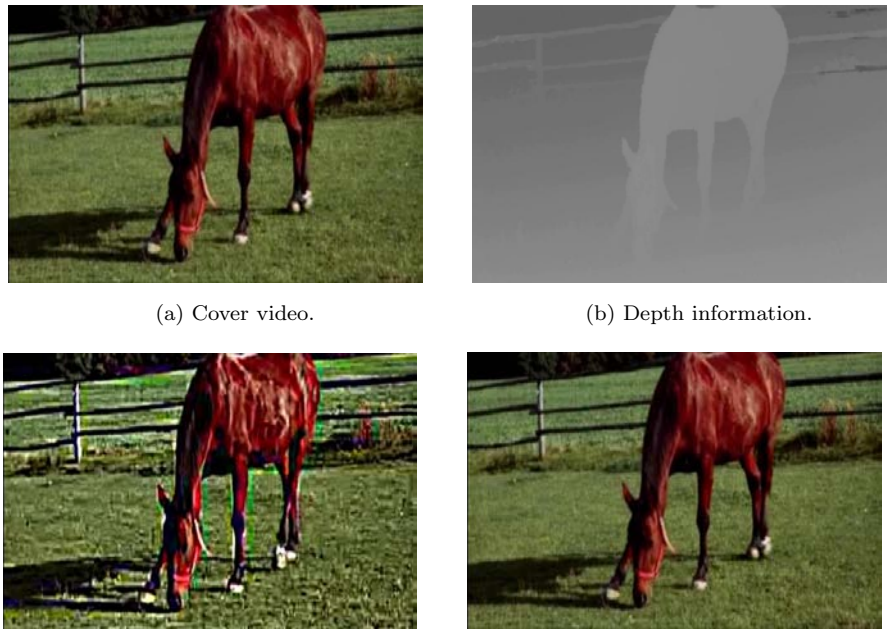
(c) Watermarked video. (d) Restored video.

Figure 4.6: Watermark embedding for video hand.

4.3 Summary

In this chapter, we first presented the experimental results on JPEG images to show that our proposed watermarking method can satisfy the capacity requirement to hide the depth information. And besides, the original cover image can be retrieved completely after the watermark is extracted, which promises a high visual quality after a large amount of watermark is embedded into the cover image.

Then we further test our watermarking scheme to the MPEG-2 compressed videos. The compression scheme is quite similar between JPEG compression and MPEG-2 compression, especially the procedures of DCT transformation and quantization, on which our watermarking method relies, are used in both of these two compression methods. In this case, a compressed depth video can be hidden into its corresponding video and the watermarked cover video can also be restored after the depth information

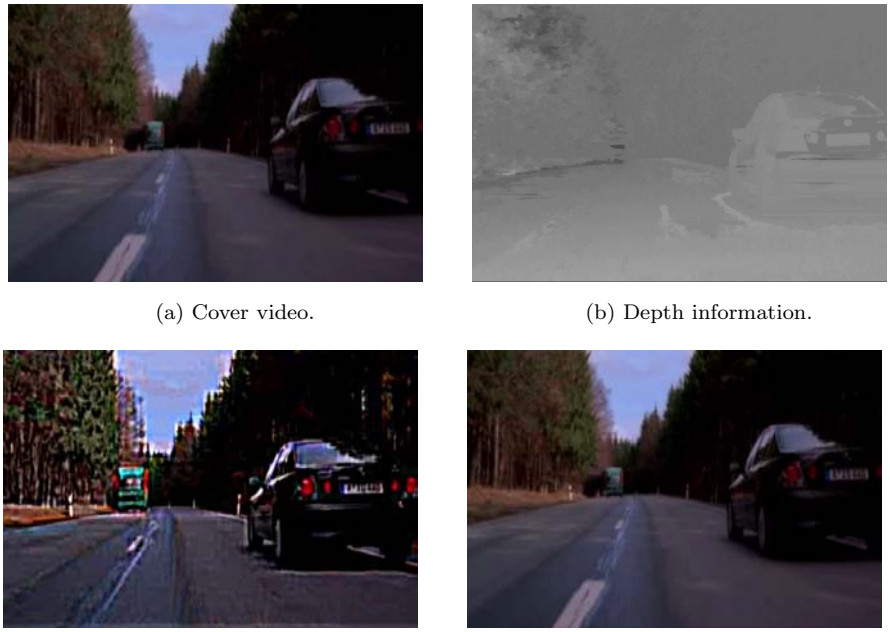


(a) Cover video. (b) Depth information.

(c) Watermarked video. (d) Restored video.

Figure 4.7: Watermark embedding for video horse.

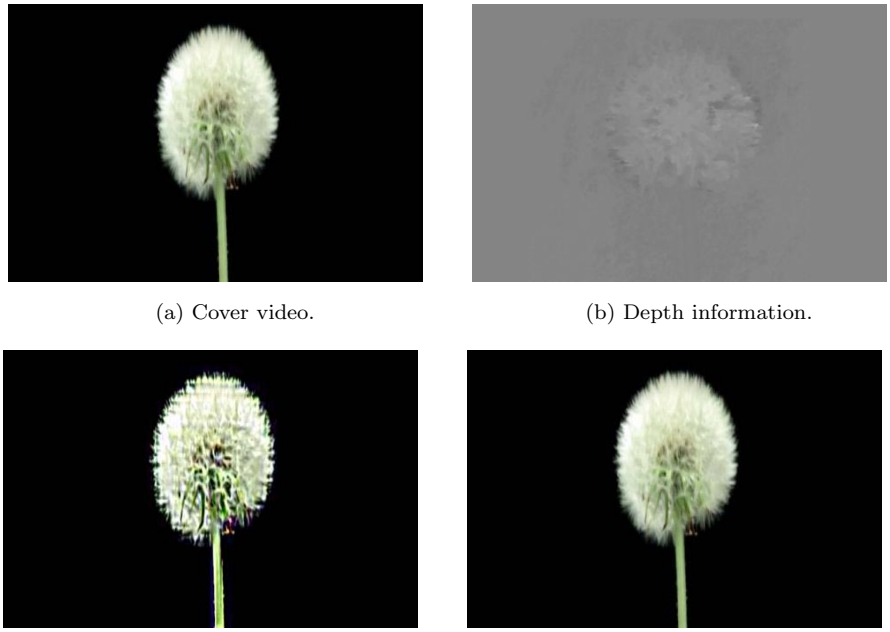
is extracted.



(a) Cover video. (b) Depth information.

(c) Watermarked video. (d) Restored video.

Figure 4.8: Watermark embedding for video car.



(a) Cover video. (b) Depth information.

(c) Watermarked video. (d) Restored video.

Figure 4.9: Watermark embedding for video flower.

Chapter 5

Conclusions and future work

High capacity and anti-compression are the two conflict characteristics for digital watermarking. In order to hide a depth map into a JPEG format image, the requirement for capacity and robustness are both very high. For conventional watermarking methods, it is very difficult to obtain both of the high watermark capacity and robustness to JPEG and MPEG-2 compression. In our scheme, a fragile reversible watermarking is applied to the quantized DCT domain and this make our watermarking scheme compatible with the digital content with JPEG and MPEG-2 format.

Since the cover content is in a compressed format (JPEG or MPEG-2), the size of the watermarked image in our scheme is significantly decreased compared to existing approaches which embedded the depth map into the un-compressed spatial domain. To further reduce the influence of the watermarking process on the size of the JPEG compressed image, the Huffman table customization and bit-shifting are utilized. As the result, the size of the watermarked image is almost equal to the sum of the size of the original JPEG image plus the size of the watermark embedded (compressed depth map). In this case, we successfully embedded the depth map into the JPEG compressed

image without introducing noticeable extra size and the cover image can be completely restored after the depth map is extracted at the receiver side. And besides, the watermarked image can be re-compressed by the process of watermark extraction, original image restoration, watermark embedding and re-compression. As to the watermarking for video, the size of watermarked video even do not change significantly, the reason for this characteristic is that the bit-rate control system in MPEG-2 system prevents the significant size change with the sacrificing of video quality.

The test results demonstrated that the scheme has sufficient capacity to hide the depth information into the host digital contend such as JPEG image and MPEG-2 video. And the requirement for high visual quality can also be satisfied since that the distortion from the reversible watermarking to the cover content can be removed completely after the detection and extraction of the watermark information.

As for our future work, we will improve our watermark embedding system for a better visual quality and smaller size increase. For the video watermarking, it is necessary to implement our watermarking scheme without the bit-rate-control for the reason that the video quality will be degraded irreversibly with the watermarking embedding and bit-rate-control. However, the bit-rate control system can be investigated carefully to reach a balance between the visual quality and size increase.

References

[1] K. Fliegel, Advances in 3D imaging systems: Are you ready to buy a new 3D TV set?, International Conference on Radioelektronika (RADIOELEKTRONIKA), pp. 1–6, April 2010.

[2] L. Onural, A. Smolic, T. Sikora, M. R. Civanlar, J. Ostermann, and J. Watson, An assessment of 3D-TV technologies, Conference on NAB Broadcast Engineering, pp. 456–467, 2006.

[3] L. Meesters, W. A. Ijsselsteijn and P. J. H. Seuntiens, A survey of perceptual evaluations and requirements of three-dimensional TV, IEEE Transanction on CSVT, Vol. 14, No. 2, pp. 381–391, 2004.

[4] A. Smolic, and D. McCutchen, 3D-TV exploration of video-based rendering technology in MPEG, IEEE Transanction on Circuits and Systems for Video Technology, Vol. 14, No. 3, pp. 348–356, 2004.

[5] K. Mueller, A. Smolic M. Kautzner, P. Eisert, and T. Wiegand, Predictive Compression of Dynamic 3D Meshes, IEEE Proceedings of International Conferenceon Image Processing (ICIP), pp. 11–14, Genova, Italy, September 2005.

[6] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis and A. Koz, Coding algorithms for 3D TV- a survey, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 11, pp. 1606–1621, 2007.

[7] L. Zhang and W.J. Tam, Stereoscopic image generation based on depth image for 3D TV, IEEE Transactions on Broadcast, Vol. 51, No. 2, pp. 191–199, 2005.

[8] D. Coltuc, On stereo embedding by reversible watermarking, International Symposium on Signals, Circuits and Systems ISSCS07, pp. 1–4, Iasi, Romania, July 2007.

[9] D. Coltuc and I. Caciula, On stereo embedding by reversible watermarking: Further Results, International Symposium on Signals, Circuits and Systems ISSCS09, pp. 1–4, Iasi, Romania, July 2009.

[10] D. Coltuc, I. Caciula and H. Coanda, Color stereo embedding by reversible watermarking, International Symposium on Electrical and Electronics Engineering (ISEEE) ISSCS09, pp. 256–259, Targoviste, Romania, September 2010.

[11] http://vision.middlebury.edu/stereo (Last visited on 9/05/2011).

[12] J.N.Ellinas, Reversible watermarking on stereo image sequences, International Journal of Signal Processing, Vol. 5, No. 3, pp. 210–215, 2009.

[13] A. Khan, A. Ali, M.T. Mahmood, I. Usman and T.S. Choi, Variable Threshold Based Reversible Watermarking: Hiding Depth Maps,

IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, pp. 59–64, October 2008.

[14] A. Khan, M.T. Mahmood, A. Ali, I. Usman and T.S. Choi, Hiding depth map of an object in its 2D image: Reversible watermarking for 3D cameras, International Conference on Consumer Electronics ICCE '09, pp. 1–2, Janerary 2009.

[15] M. Swanson, B. Zhu, and A. Tewfik, Data hiding for video in video, IEEE International Conference on Image Processing, Vol. 2, pp. 676–679, October 1997.

[16] M. Swanson, B. Zhu, and A. Tewfik, Robust data hiding for images, IEEE Workshop Proceedings on Digital Signal Processing, pp. 37–40, September 1996.

[17] L.Tse-Hua and A.H. Tewfik, A novel high-capacity data-embedding system, IEEE Transactions on Image Processing, Vol. 15, No. 8, pp. 2431–2440, 2006.

[18] N.K. Kalantari and S.M. Ahadi, A Logarithmic Quantization Index Modulation for Perceptually Better Data Hiding, IEEE Transactions on Image Processing, Vol. 19, No. 6, pp. 1504–1517, 2010.

[19] H.Y. Shum, S.B. Kang and S.C. Chan, Survey of Image-based representations and compression technique, IEEE Transactions on Circuits system and Video Technology, Vol. 13, No. 11, pp. 1020–1037, 2003.

[20] P. Zanuttigh and G.M. Cortelazzo, Compression of depth information for

3D rendering, 3DTV Conference:The True Vision C Capture, Transmission and Display of 3D Video, pp. 1–4, May 2009.

[21] K.J. Oh, S. Yea, A. Vetro and Y.S. Ho, Depth reconstruction filter for depth coding, Electronics Letters, pp. 305–306, 2009.

[22] T. Kalker and F.M.J. Willems, Capacity bounds and constructions for reversible data-hiding, International Conference on Digital Signal Processing , pp. 71–76, Vol. 1, 2003.

[23] M. Barni, F. Bartolini, A. De Rosa, and A. Piva, Capacity of full frame DCT image watermarks, IEEE Transancation on Image Processing, Vol. 9, pp. 1450–1455, August 2000.

[24] M. Ramkumar and A. N. Akansu, Information theoretic bounds for data hiding in compressed images, IEEE Proceedings of 2nd Workshop on Multimedia Signal Processing, pp. 566–569, 1998.

[25] S. D. Servetto, C. I. Podilchuk and K. Ramchandran, Capacity issues in digital image watermarking, IEEE International Conference on Image Processing, Vol. 1, pp. 445–449, Chicago, Illinois, USA, October 1998.

[26] R. Sugihara, Practical capacity of digital watermark as constrained by reliability, IEEE International Conference on Information Technology: Coding and Computing, pp. 85–89, April 2001.

[27] D.B. Graziosi, N.M.M. Rodrigues, C.L. Pagliari, S.M.M. de Faria, E.A.B. da Silva and M.B. De Carvalho, Compressing depth maps using multiscale recurrent pattern image coding, Electronics Letters, pp. 340–341, 2010.

[28] J.B. Feng, I.C. Tsai and Y.P. Chu, Reversible watermarking: current status and key issues, International Journal of Network Security, Vol. 2, No. 3, pp. 161–171, May 2006.

[29] M.U. Celik, G. Sharma, A.M. Tekalp and E. Saber, Reversible data hiding, International Conference on Image Processing, pp. 157–160, NY, USA, September 2002.

[30] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, Localized lossless authentication watermark (LAW), International Society for Optical Engineering, Vol. 50, No. 20, pp. 689–698, California, USA, Janurary 2003.

[31] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, Lossless generalized-LSB data embedding, IEEE Transanction on Image Processing, Vol. 14, No. 2, pp. 253–266, 2005.

[32] A.M. Alattar, Reversible watermark using difference expansion of quads, IEEE Processings of International Conference on Acoustics, Speech, and Signal , Vol. 3, pp. 377–380, Montreal, Canada, May 2004.

[33] A.M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, IEEE Transanction on Image Processing, Vol. 13, pp. 1147–1156, 2004.

[34] A.M. Alattar, Reversible watermark using difference expansion of triplets, IEEE International Conference on Image Processing, Vol. 1 pp. 501–504, Catalonia, Spain, Septtember 2003.

[35] Jun Tian, Reversible data embedding using a difference expansion, IEEE

Transanction on Circuit and Systems for Video Technology, Vol. 13, pp. 890–896, 2003.

[36] D.M. Thodi, J.J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Transanction on Image Processing, Vol. 16, pp. 721–730, 2007.

[37] Sunil Lee and C.D. Yoo, Reversible Image watermarking based on integer-to-integer wavelet transform, IEEE Transanction on Information Forensics and Security, Vol. 2, pp. 321–330, 2007.

[38] V. Sachnev, Hyoung Joong Kim, Jeho Nam, Suresh and Yun Qing Shi, Reversible watermarking algorithm using sorting and prediction, IEEE Transanction on Circuit and systems for video Technology, Vol. 19, pp. 989–999, 2009.

[39] Xiang Wang, Xiaolong Li, Bin Yang, Zongming Guo, Efficient generalized integer transform for reversible watermarking, IEEE Signal Processing Letters, Vol. 17, pp. 567–570, 2010.

[40] Shaowei Weng, Yao Zhao, Jeng-Shyyang Pan and Rongrong Ni, Reversible watermarking based on invariability and adjustment on pixel pairs, IEEE Signal Processing Letters, Vol. 15, pp. 721–724, 2008.

[41] Yongjian Hu, Heung-Kyu Lee, Kaiying Chen and Jianwei Li, Difference expansion based reversible data hiding using two embedding directions, IEEE Transanction on Multimedia, Vol. 10, pp. 1500–1512, 2008.

[42] C.C. Chang, C.C.Chen and Y.H., Reversible data-embedding scheme using

differences between original and predicted pixel values, IET Information Security, Vol. 2, pp. 35–46, 2008.

[43] Wei-Liang Tai, Chia-Ming Yeh and Chin-Chen Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Transanction on Circuits and Systems for Video Technology, Vol. 19, pp. 906–910, 2009.

[44] C. D. Vleeschouwer, J. E. Delaigle, and B. Macq, Circular interpretation of bijective transformations in lossless watermarking for media asset management, IEEE Transanction on Multimedia, Vol. 5, No. 1, pp. 97–105, 2003.

[45] C. D. Vleeschouwer, J. E. Delaigle, and B. Macq, Circular interpretation of histogram for reversible watermarking, IEEE Proceedings of 4th Workshop on Multimedia Signal Processing, pp. 345–350, 2001.

[46] E. Y. Lam and J. W. Goodman, A mathematical analysis of the dct coefficient distributions for images, IEEE Transanction on Image Processing, Vol. 9, No. 10, pp. 1661–1666, 2000.

[47] B. Yang, M. Schmucker, X. Niu, C. Busch, and S. Sun, Interger DCT based reversible image watermarking by adaptive coefficient modification, SPIE Proceeedings on Security Steganography and Watermarking of Multimedia Contents, pp. 218–229, San Jose, California, USA, March 2005.

[48] http://sp.cs.tut.fi/mobile3dtv/video-plus-depth/ (Last visited on 02/09/2011)