國立交通大學

多媒體工程研究所

碩士論文

利用 3D KINECT 影像做資訊隱藏並應用於影像驗證、秘密傳輸及版權保護

Data Hiding via 3D KINECT Images and Its Applications for Authentication, Covert Communication, and Copyright Protection

研 究 生：廖為帥

指導教授：蔡文祥　教授

中華民國一百零二年六月

利用 3D KINECT 影像做資訊隱藏並應用於影像驗證、秘密傳輸及版權保護

Data Hiding via 3D KINECT Images and Its Applications for Authentication, Covert Communication, and Copyright Protection

研究生：廖為帥　　Student: Wei-Shuai Liao

指導教授：蔡文祥　　Advisor: Prof. Wen-Hsiang Tsai

國立交通大學
多媒體工程研究所
碩士論文

A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年六月

Data Hiding via 3D KINECT Images and Its Applications for Authentication, Covert Communication, and Copyright Protection

Student : Wei-Shuai Liao　　　Advisor: Wen-Hsiang Tsai

Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

With the growing popularity of the Internet and KINECT devices, related applications of KINECT devices are increasing in various fields and the images acquired by KINECT devices are more commonly seen in people's daily life. Protection of the 3D image data which are acquired by KINECT devices becomes an important issue. In this study, we propose three data hiding methods for authentication and copyright protection of KINECT images (including the depth and color images), as well as for covert communication via them.

For authentication, a data hiding method for authenticating KINECT images by embedding authentication signals into the depth and color images acquired with the KINECT device is proposed. The proposed method utilizes the features of KINECT images and the ranges of the pixel values in them to achieve the goal of authenticating the color and depth images as a whole.

For cover communication, a data hiding method for this purpose via KINECT images is proposed, which utilizes depth holes in the depth image acquired by the KINECT device as well as a new interpolation technique to hide secret messages in the depth image.

For protection of the copyright of KINECT images, a method of 3D visible watermarking is proposed for embedding 3D visible watermarks into 3D images, which results from combining depth and color images acquired by the KINECT device. In addition, embedding the 3D visible watermark will cover some regions of the original 3D image. This problem is solved by transforming the original data of the regions into a recovery data sequence, which is then embedded into the watermarked 3D image by two schemes, one being a difference expansion scheme and the other a reversible contrast mapping scheme. In order to prevent malicious users from extracting the recovery information, the recovery data sequence and the embedding locations are both randomized by a secret key.

Good experimental results are also presented to show the feasibility of the proposed methods for the related applications.

利用 3D KINECT 影像做資訊隱藏並應用於影像驗證、秘密傳輸及版權保護

研究生：廖為帥　　　　指導教授：蔡文祥　博士

國立交通大學多媒體工程研究所

摘要

隨著網路及 KINECT 裝置的普及，在各個領域中與 KINECT 相關的應用也隨著增加，而在人們日常生活中藉由 KINECT 裝置所擷取到的影像也更為常見。因此保護這些經由 KINECT 裝置所獲取的 3D 影像資料已成為一個重要的議題。此研究針對 KINECT 影像(包含深度和彩色影像)提出三個資訊隱藏的方法，來達到影像驗證、祕密傳輸及版權保護的目的。

在影像驗證的方面，本研究提出了一個資訊隱藏的方法，該方法是基於 KINECT 影像的特性和其像素值範圍，藉由嵌入驗證訊號到影像中的方式，來達到同時驗證 KINECT 深度及彩色影像的目的。

在祕密傳輸方面，本研究使用 KINECT 影像提出了一個資訊隱藏的方法，該方法利用 KINECT 深度影像中的「深洞」(“depth holes”)並使用一新內插技術，將祕密訊息藏入深度影像中並保持影像品質不變。

在 KINECT 影像的版權保護方面，本研究提出了一個直入 3D 可視浮水印的方法，該方法嵌入一個 3D 可視浮水印到一個由 KINECT 深度和彩色影像合成的 3D 影像之中，嵌入 3D 可視浮水印時會覆蓋掉原始 3D 影像中的某些區域；為此，本研究將這些區域的原始資料轉換成一種回復資訊，並使用差值擴張(difference expansion)和反轉式對比應對(reversible contrast mapping)兩方法將該資訊嵌入到具有浮水印的 3D 影像中。為了防止有心人士取得回復資訊，本研究使用一把祕密鑰匙，將回復資訊的順序與所嵌入的位置之順序打亂。最後，上述方法經由實

驗驗證結果良好，顯示出本研究所提方法確實可行。

ACKNOWLEDGEMENTS

The author is in hearty appreciation of the continuous guidance, discussions, and support from his advisor, Dr. Wen-Hsiang Tsai, not only in the development of this thesis, but also in every aspect of his personal growth.

Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Institute of Computer Science and Engineering at National Chiao Tung University for their suggestions and help during his thesis study.

Finally, the author also extends his profound thanks to his dear mom and dad for their lasting love, care, and encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Motivation

With the rapid development of computer network technologies, the Internet becomes popular and used widely in people’s daily life nowadays. Through the Internet, we can not only conduct communication with people but also share information. Too many kinds of information can be found on the Internet, such as chatting contents, photo images, intelligence data, etc. Certain important information is private and should be protected because malicious hackers might want to steal or tamper with such information on the Internet. Therefore, how to protect different types of information and check the correctness of them has become important in people’s life.

In addition, with the fast advance of computer vision technologies, 3D techniques become more and more popular. Many devices are invented to capture 3D information, such as the KINECT sensor provided by Microsoft. Because of the ubiquity of KINECT sensors, even nontechnical people can easily get 3D information in their daily life. Because 3D image data contains more information than 2D ones, we can check in more detail the spatial relation between any two objects in the real world via the use of the 3D data captured by 3D imaging devices. 3D image data usually contains depth information that is very helpful for various purposes in our daily life. For example, in the application of video surveillance, we can get easily from 3D image data information like the height of an intruding man, the distance between two objects in an event, and so on. Because of these characteristics and usefulness of 3D image data, the protection of 3D image data in various applications becomes an important work.

In this study, we propose a method for authentication of KINECT images (including the depth and color images) to verify their correctness in order to prevent the images from being tampered with by malicious persons. Besides, due to the popularity of the Internet, image data are shared and transferred frequently on the Internet; therefore, we also propose a method for covert communication via KINECT images, by which a user can send secret messages to persons he/she want without being intercepted or tampered with. In addition, to protect the copyright of KINECT images, we propose as well a method of 3D visible watermarking. The details of these proposed methods will be described in detail in subsequent sections or chapters.

1.2 General Review of Related Works

The invention of the Microsoft KINECT sensor [1] brings a lot of effects and opportunities not only in multimedia computing, but also in the gaming industry. The Kinect sensor changes the way by which people communicate with computers or games, i.e., changes the human interfacing technique. With the Kinect sensor, people can use just the body language to make the computer "understand" what the user is doing. The key change in this kind of user experience created by the Kinect sensor is the "third-dimension detection" technique that enables the computer to "realize" the user's movement as well as the environment condition.

Because of the popularity of the KINECT sensor, acquisitions of 3D image data are getting easier and easier. The 3D information captured by the KINECT sensor includes both depth and color images, with the former showing the distance from the KINECT sensor to the target object, and the color image being just like a commonly-seen RGB image. We call either the depth image or the color image a *KINECT image*. The more information is included in KINECT images, the more important the protection of the security of KINECT images is. Therefore, in this study we try to design information hiding techniques for the purpose of protecting KINECT images.

The hardware details of the KINECT sensor and the principle of using it to acquire depth image data will be introduced in Section 2.1. In Section 2.2, we will review previous studies of the structures of the depth and color images taken by the KINECT sensor.

Besides, many techniques for embedding fragile authentication signals for the purpose of image authentication have been proposed in the past. Specifically, the studies of [2-4] authenticate images at the pixel level such that any tampered image part can be identified pixel by pixel, yielding detailed tampering localization results. We will review the details of these studies in the Section 2.3. Furthermore, many data hiding techniques via images [5-12] have been proposed and are widely used in a lot of applications, for instance, covert communication, copyright protection, etc. Some techniques among them can be employed to conduct losslessly reversible data embedding and extraction, like those in [6-10]. As to the application of covert communication, a method of combining a new data hiding technique with a secret sharing scheme [13] is proposed in this study. Before that, we will also give a detailed review of existing related methods in Section 2.4. In Section 2.5, some methods of visible watermarking for images proposed in the past will be reviewed as well.

1.3 Overview of Proposed Methods

1.3.1 Terminologies

The definitions of some related terminologies used in this study are described as follows.

1. *Secret*: a secret (or a secret message) is a piece of information that is important and should be preserved properly and not revealed to unauthorized people.
2. *Secret share*: a secret share is generated from a given secret message according to the Shamir secret sharing scheme [3].
3. *Image authentication*: image authentication is a process for verifying the integrity and fidelity of an image by checking the authentication signals embedded in it.
4. *Protected image*: a protected image is one into which some authentication signals are embedded for the purpose of image authentication.
5. *Embedding process*: an embedding process is the process of embedding some kind of data (e.g., secret messages) into an image.
6. *Extraction process*: an extraction process is the process of extracting the hidden data from an image.

1.3.2 Brief Descriptions of Proposed Methods

In this study, we propose several methods for data hiding via KINECT images and their applications. These methods are briefly described in the following.

(A) Authentication of KINECT images by data hiding techniques —

A data hiding technique for authentication of KINECT images by embedding authentication signals into the depth and color images captured with the KINECT sensor is proposed in this study. Specifically, we try to achieve the goal of authenticating both the depth and color images *together* by authenticating either of the depth and color images using the data of the other. That is, we use the depth image to generate authentication signals and embedding them into the color image; and reversely use the color image to generate authentication signals and embedding them into the depth image. To enhance the security of the proposed method, we select random positions in the depth or color images to embed authentication signals in order to reduce the possibility for a malicious user to figure out the locations of the embedded authentication signals in the KINECT images.

(B) Covert communication via KINECT images by interpolation at depth holes —

A data hiding method for covert communication via KINECT images by interpolation at *depth holes* is proposed in this study, where depth holes mean those 3D locations where the KINECT sensor does not provide depth data due to hardware or environment limitations. At first, a secret sharing scheme [3] is used to generate secret shares from a given secret message. Then, the generated secret shares are hidden, by a new technique of interpolation proposed in this study, at the depth holes in the depth image captured with the KINECT camera. Therefore, the proposed method can enhance the quality of the depth image, in addition to hiding the secret message into the KINECT image.

(C) Copyright protection of KINECT images by 3D visible watermarking —

A method of using data hiding techniques to protect the copyright of KINECT images by 3D visible watermarking is proposed in this study. In embedding process, at first the depth and color images are combined into a single *3D image*. Then, a given 3D visible watermark is embedded into the generated 3D image simply by *data replacement*. To recover the replaced original 3D image data, recovery information should be saved. For this, two schemes are proposed. One is to hide the recovery information in the 3D image by a difference expansion scheme and the other is to use the reversible contrast mapping scheme. Afterwards during the stage of watermark removal, the recovery information is extracted first. Then, the embedded 3D visible watermark is eliminated from the 3D image. And finally the replaced original 3D image data is recovered by using the extracted recovery information. In this way, the copyright of KINECT images can be protected.

1.4 Contributions

New methods for data hiding and applications via 3D KINECT images are proposed in this study. The contributions made in this study are listed in the following.

1. Three new applications involving KINECT images (including depth and color images) are proposed.
2. A data hiding technique using KINECT image features based on Lee and Tsai’s image authentication technique [2] is proposed.
3. A method of using the features of the depth image to authenticate the color image and using those of the color image to authenticate the depth image is proposed.
4. A data hiding technique in KINECT depth images by interpolation at depth holes is proposed.

5. A method of covert communication using the above technique of interpolation at depth holes is proposed.
6. Two techniques using difference expansion and reversible contrast mapping for hiding data into 3D images constructed from KINECT images are proposed.
7. A method for copyright protection of KINECT images by 3D visible watermarking using the above two data hiding techniques is proposed.

1.5 Thesis Organization

The remainder of this thesis is organized in following descriptions. An introduction to the Kinect sensor as well as the structures of image data captured by the Kinect sensor, and reviews of the applications and methods of image authentication, covert communication via images, and visible watermarking in images will be described in Chapter 2. The proposed method for authentication of KINECT images is described in Chapter 3. The proposed method for covert communication via KINECT images by interpolation at depth holes is described in Chapter 4. In Chapter 5, the proposed method for copyright protection of KINECT images by 3D visible watermarking is described. Finally, conclusions and some suggestions for future works are included in Chapter 6.

Chapter 2
Review of Related Works and KINECT Image Structures

In this chapter, we will give a review of the KINECT sensor and the structures of the depth and color images taken by the KINECT device in Sections 2.1 and 2.2, respectively. And then we will give also a review of the existing data hiding techniques for image authentication, covert communication via images, and copyright protection of images, in Sections 2.3 through 2.5, respectively.

2.1 Previous Studies of 3D Image Acquisition Using KINECT Devices

The release of the Microsoft Kinect sensor was probably one of the biggest impacts in the research field of computer vision. Kinect sensors have created many opportunities for multimedia computing. In this section, we give a review of the hardware of the KINECT sensor and the principle of the operations which can be conducted using the KINECT sensor.

The Kinect sensor includes a color VGA video camera, a depth sensor, a multi-array microphone, and a tilt motor for sensor operations and adjustments. The horizontal field of view of the KINECT device is 57 degrees, the vertical field of view is 43 degrees, and the physical tilt range is ± 27 degrees. The color camera aids in detecting three color components: red, green and blue, like other cameras. The main difference between commonly-seen cameras and the Kinect sensor is that the Kinect device has an extra depth detection sensor. The depth detection sensor is composed of an infrared projector and a monochrome Complementary Metal-Oxide Semiconductor (CMOS) sensor, which work together to capture the distance information between the depth sensor and the objects in front of the Kinect device. The reason why the Kinect sensor can acquire the depth information comes from the use of the PrimeSense’s light coding-patented technology. The light coding technology works by coding the scene with near-IR light, which is invisible to the human eye, and then uses the CMOS image sensor and the chips to execute sophisticated parallel computational algorithms to decipher the received light-coding infrared patterns to produce a VGA-size depth image of the scene.

The key of the mentioned light coding technique is the use of laser speckles — when the laser is projected on objects with rough surfaces or through the frosted glasses, it will generate random reflecting speckles, integrally called the speckle pattern. The speckle pattern is highly randomized and the pattern image changes with different distances. The speckle pattern images that are captured by the Kinect sensor in any two places of the real space are different. According to speckle patterns, all the places in the real space can be marked. Then, the depth information can be obtained by decoding the laser speckle pattern on the object.

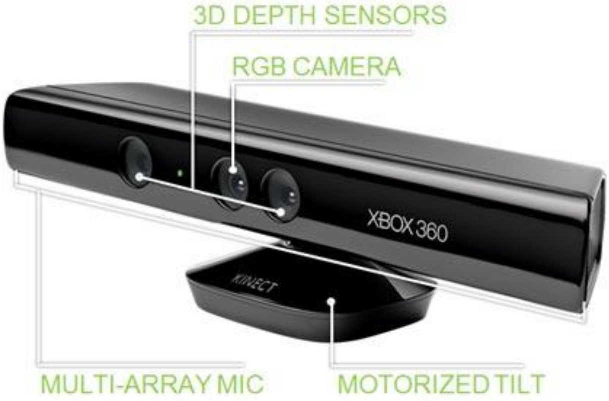


Figure 2.1 Hardware of the KINECT device.

2.2 Previous Studies of Structures of Depth and Color Images Taken by KINECT Devices

In this section, we will give a review of the functions of the KINECT sensor and the structures of the depth and color images taken by the KINECT sensor. The KINECT sensor brings many effects in not only various fields of research but also our daily life. Because of the popularity of the KINECT sensor, people can use it easily to develop various kinds of applications, for example, detection of the positions of the user’s hands for browsing of websites according to the user’s movements only.

Many developers use the Open Natural Interaction (OpenNI) software development kit (SDK) of the KINECT sensor to develop related applications. The OpenNI framework is an open-source SDK useful for development of 3D sensing middleware libraries and applications. In addition, there are other SDKs like the Kinect-for-Windows SDK, which is also used by many developers for R & D. In this study, we use the OpenNI SDK to acquire data with the KINECT sensor. The acquired data include depth and color information, with the color information being like a commonly-seen color image with resolution 640×480; and the depth information being a range image also with resolution 640×480. The depth range provided by the KINECT sensor using the Kinect-for-Windows SDK is from 800mm to 4000mm, but that provided by the use of the OpenNI SDK is up to is the maximum of about 8000mm. As the depth distance detected by the KINECT sensor increases, more depth data will be missed in the detected result. For example, the missing depth value start from 611mm, and then 622mm, 631mm, 638mm, etc. When the depth value comes to 7960mm, the values of 7961mm to 8146mm are missing, so the inaccuracy of the detected value is 185mm. According to this phenomenon, we can realize that the larger the depth value captured by the KINECT sensor, the lower accuracy the detected depth values have. If a user wants to interact with the KINECT sensor, the better range of distances between the KINECT sensor and the user is from 1200mm to 3600mm, which is advised by the KINECT development official website.

The acquired data by the KINECT sensor includes depth and color images as mentioned previously, and either of the depth and color images will be called a KINECT image in this study.

2.3 Review of Techniques for Image Authentication

With the advance of the computer and Internet technologies, security of digital image data is considered as a significant issue today. Thus, many techniques for embedding authentication signals for the purpose of image authentication have been proposed in the past. Specifically, the methods proposed in [2-4] authenticate images at the pixel level such that any pixel of a tampered image part can be identified, and then the result of detailed tampering localization is reported.

The method proposed in Liu et al. [2] generates a binary image that is mapped from the difference image computed from the cover image and its so-called chaotic pattern. And the least-significant-bit (LSB) plane is used to accommodate the binary image as a fragile watermark for use in later image authentication.

In Lee and Tsai [3, 4], a grayscale image authentication method was proposed. The method is based on a bin-mapping scheme which divides the grayscale range into two parts, the five MSBs and the remaining three LSBs. The former is used to generate a 3-bit bin code as the authentication signal for each pixel in the input cover image. Then, the authentication signals are embedded randomly into the other pixels of the image.

The authentication signals are utilized not only for detecting and localizing tampered pixels but also for generating representative values for repairing the tampered pixels.

2.4 Review of Techniques for Covert Communication via Images

Many data hiding techniques have been proposed for various purposes such as authentication and covert communication in the past. To achieve the goal of hiding the data imperceptibly, data hiding utilizing the weaknesses of the human's vision system have been investigated intensively. A widely known method is least significant bit (LSB) modification which changes the LSBs of the pixel value in an image to embed information. For example, Chan and Cheng [5] presented a data hiding method by simple LSB substitution. On the other hand, data hiding techniques using pairs of image pixels to hide information have also been proposed, like Tian [6] who proposed a reversible data embedding method by using a difference expansion scheme which is based on simple reversible integer transformations. This method calculates the differences of neighboring pixel values, and selects some difference values of the pixel pairs for the difference expansion. Then, the information is embedded into the expanded differences of the pixel pairs. Since the modified values are generated from the differences between manipulated pixel pairs, the original pixel values can be recovered easily.

2.5 Review of Techniques for Copyright Protection of Images

Because of the popularity of the Internet, acquisitions of digital information from the network becomes easier and easier nowadays. Thus, protection of the copyright of digital information on the Internet is more important than ever before. About the topic of copyright protection of images, digital watermarking has been used widely in a lot of applications [7-12]. Digital watermarking means embedding some kinds of information, like ownership information, company logo, etc., into digital images that should be protected.

In general, digital watermarking techniques for images can be categorized into two types: visible and invisible. The first type of technique, visible watermarking, is to embed clearly visible marks into images. The embedded visible watermark is usually irremovable and leaves permanent distortion to the original image. The technique of the second type aims to embed the copyright information imperceptibly into images so that in case of copyright infringement, the hidden information can be retrieved to identify the ownership of the protected host image.

Both visible and invisible watermarking techniques yield the distortion of the host image after the embedding process. A group of techniques, named *reversible* watermarking [7-10], allow authorized users to remove the embedded watermark and save the original content of images as needed. Besides, some reversible watermarking techniques guarantee *lossless image recovery*, which means that the recovered image is identical to the original image. The techniques of lossless recovery is important in some applications , for example, images for military uses, historical art imaging, or related applications of medical image analysis. In these applications, any permanent distortion generated by watermarking is not allowed.

In [7], Alattar extended Tian’s algorithm [6] to utilize the difference expansion of vectors, instead of pairs of pixels, to increase the hiding ability and the computation efficiency.

In [8], Coltuc and Chassery presented a high-capacity data embedding scheme without using any additional data compression operation. This scheme is based on a reversible contrast mapping (RCM) technique, which is a simple integer transform defined on pairs of pixels. It partitions the cover image into pairs of pixels and divides the pairs into three groups, and then conducts a respective embedding process on each pair group. This RCM scheme provides almost similar embedding bit-rates when compared to the difference expansion approach, while it has a considerably lower mathematical complexity.

Chapter 3

Authentication of KINECT Images by Data Hiding Technique

3.1 Introduction

Because of the growing popularity of the KINECT device, the depth and color images, also called KINECT images in this study, acquired by KINECT devices are more commonly seen in various applications. Like other digital images, the KINECT images also need be verified for their correctness in order to prevent them from being tampered with by malicious persons when they are transmitted or kept in storages. For this reason, we propose a method for authentication of KINECT images in this study. The detail of the method will be described in this chapter.

First, the definition of the problem and the idea of the proposed method are given in Section 3.1. In Section 3.2, we will describe the process of generating authentication signals and embedding them into the KINECT images. Then, in Section 3.3, the process of extracting the embedded authentication signals from the protected KINECT images will be described. In Section 3.4, the recovery of the original depth and color images, and repairing of the possibly tampered versions of them will be described. Experimental results showing the feasibility of the method are given in Section 3.5. Finally, some discussions and a brief summary are given in the last section of this chapter.

3.1.1 Problem Definition

As 3D sensing technologies are growing vigorously, related applications of the KINECT sensor are also increasing in various fields. Since the KINECT sensor can be used to acquire the depth information, the data can be used more extensively in applications than the 2D data can, for example, as the 3D information of marble sculptures created by famous artists and the models of historical architectures or objects encountered in daily life. These kinds of information are often very important, so the correctness of them must be guaranteed.

The range of the depth values that are provided by the KINECT device is different from that of the general values of color-image pixels. In addition, the pair of the depth and color images acquired by the KINECT device at each identical instant should be protected together to keep their relation in time unchanged. No matter whether the color or the depth image is tampered with by malicious persons or not, the tampered region should be detected by an authentication method, and better be repaired. These requirements should be considered when designing an authentication method for KINECT images, as is done in this study.

3.1.2 Proposed Ideas

The major idea of the proposed authentication method for KINECT images was inspired by the concept involved in the authentication method proposed by Lee and Tsai [4] as well as by some natures of KINECT image features. The proposed method aims to authenticate, as a whole, every pair of color and depth images taken by a KINECT device at an identical instant. The proposed method utilizes the features of KINECT images and the ranges of the pixel values in them to achieve the goal of authenticating them together as a whole.

As an inherent feature of KINECT images, a pixel value in the depth image is represented by a 13-bit binary string, as mentioned in Chapter 2, in order to cover the entire range of the depth values. And the value of a pixel in the color image, including the information of red, green, and blue, is represented by three 8-bit binary strings, respectively.

Furthermore, in the proposed method for KINECT image authentication, as shown in Fig. 3.1 the 13-bit depth value *D* of each pixel P_d in a depth image is divided into two parts － the nine MSBs of *D* and the three LSBs. The former is used to generate an authentication signal for the depth pixel P_d itself. The authentication signal is then embedded into five LSBs of a color-image pixel P_c randomly, where the five LSBs of P_c includes two LSBs of the red color value, two LSBs of the green color value, and one LSB of the blue color value, or we can say equivalently, the mentioned five LSBs of P_c includes the (2, 2, 1) LSBs of the (R, G, B) values of P_c. On the other hand, the nine MSBs of the color-image pixel P_c are used to generate an authentication signal for the color-image pixel P_c itself, and the signal is embedded into the three LSBs of the depth pixel P_d randomly, where the nine MSBs of P_c includes (3, 3, 3) MSBs of the (R, G, B) values of P_c. In addition, the generated signals not only can be used to verify the correctness of a pixel *P* (P_d or P_c) in the color or depth image, but also can be used to repair *part of* the value of *P* when *P* is authenticated to have been tampered with.

The detailed algorithms about the proposed methods and the related processes of authentication of KINECT images are presented in the following sections.

3.2 Generation and Embedding of Authentication Signals

In this section we will introduce the details of the implemented processes of authentication signal generation and embedding according to the proposed method. An illustration of the processes is illustrated in Figure 3.1. The detailed process of generation of authentication signals is described in Section 3.2.1, and the detailed process of embedding the generated authentication signals into KINECT images is described in Section 3.2.2.

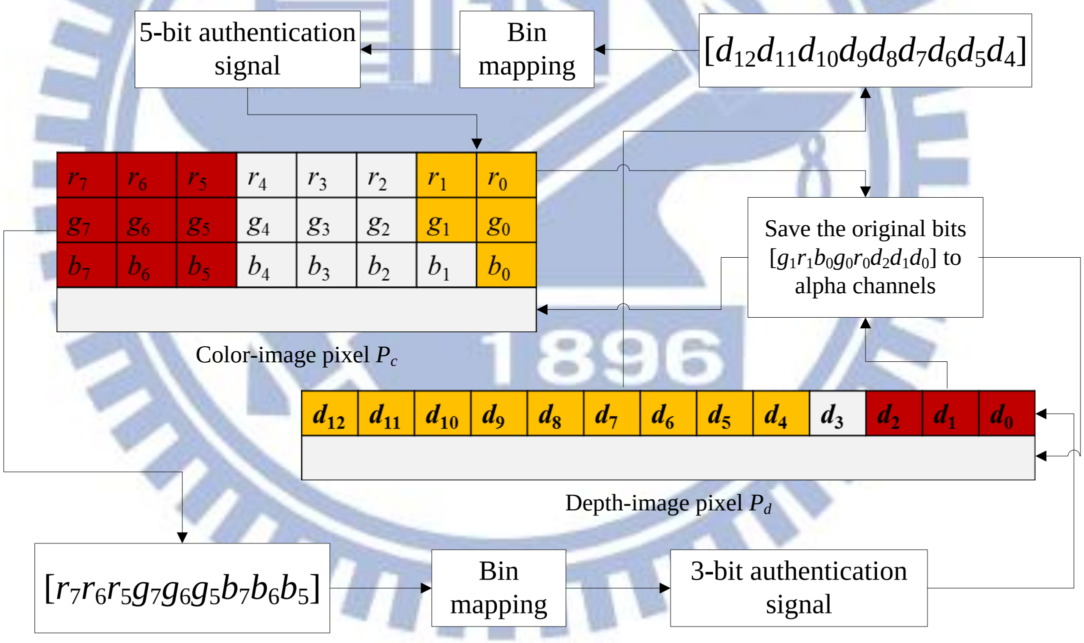

5-bit authentication signal
Bin mapping
$[d_{12}d_{11}d_{10}d_9d_8d_7d_6d_5d_4]$
r_7 r_6 r_5 r_4 r_3 r_2 r_1 r_0
g_7 g_6 g_5 g_4 g_3 g_2 g_1 g_0
b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
Save the original bits $[g_1r_1b_0g_0r_0d_2d_1d_0]$ to alpha channels
Color-image pixel P_c
d_{12} d_{11} d_{10} d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0
Depth-image pixel P_d
$[r_7r_6r_5g_7g_6g_5b_7b_6b_5]$
Bin mapping
3-bit authentication signal

Figure 3.1 Illustration of the authentication signals generation and embedding.

3.2.1 Authentication Signal Generation

In the proposed authentication signal generation process, at first we transform the value of a depth pixel P_d in the depth image into a 13-bit binary string, d_{12}, d_{11}, d_{10}, …, d_0. Then, we also transform the (R, G, B) values of each color-image pixel P_c into three 8-bit binary strings, r_7, r_6, r_5, …, r_0, g_7, g_6, g_5, …, g_0, and b_7, b_6, b_5, …, b_0, respectively. Next, we transform the nine MSBs of P_d into an integer m and the (3, 3, 3) MSBs of the (R, G, B) values of P_c as a whole are also transformed into an integer n. Then, we apply a modified version of a *bin-mapping* scheme mentioned in Lee and Tsai [4] for the purpose of compressing these MSBs information before embedding them. Additionally, we map the depth-value range specified by the nine MSBs into 32 equal-length intervals, called *bins*. And each bin is indexed by a *decimal* integer called a *bin number*, which corresponds to a 5-bit *binary* number called a *bin code*. The 32 bins and their corresponding bin numbers and bin codes are shown in Table 1. On the other hands, the color-value range specified by the nine MSBs is also mapped into eight *bins*, and the eight bins and their corresponding bin numbers and bin codes are shown in Table 2. Finally, the bin code of each pixel is taken to be the authentication signal of the pixel.

The proposed technique above for generating authentication signals using parts of depth pixel values and color pixel values is described as an algorithm, Algorithm 3.2.1, as follows.

Table 3.1 Bins, bin number, bin codes, and representative values of bins used in the generation of authentication signals for depth image pixels in this study.

Bin	Bin number	Bin code	Representative value of bin
[0, 31]	0	00000	16
[32, 63]	1	00001	48
[64, 95]	2	00010	80
⋮	⋮	⋮	⋮
[416, 447]	29	11101	432
[448, 479]	30	11110	464
[480, 511]	31	11111	496

Table 3.2 Bins, bin number, bin codes, and representative values of bins used the generation of authentication signals for color image pixels in this study.

Bin	Bin number	Bin code	Representative value of bin
[0, 63]	0	000	32
[64, 127]	1	001	96
[128, 191]	2	010	160
[192, 255]	3	011	224
[256, 319]	4	100	288
[320, 383]	5	101	352
[384, 447]	6	110	416
[448, 511]	7	111	480

Algorithm 3.2.1: authentication signals generation.

Input: a color pixel value of P_c including a red value R, a green value G and a blue value B and a depth pixel P_d with value D.

Output: the 3-bit color authentication signal s and the 5-bit depth authentication signal t.

Steps:

Step 1. Transform the depth value of P_d into thirteen bits, d_{12}, d_{11}, d_{10}, …, d_0. And transform the red color values R, G, and B into three eight-bit strings, r_7, r_6,

r_5, …, r_0, g_7, g_6, g_5, …, g_0, and b_7, b_6, b_5, …, b_0.

Step 2. Transform the nine MSBs, d_{12}, d_{11}, d_{10}, …, $d_5$into an integer m; and concatenate the three binary sub-strings, r_7, r_6, r_5, g_7, g_6, g_5, b_7, b_6, b_5, and transform the result into an integer n.

Step 3. Map the integer m into a bin indexed by a bin number B_m computed by the function $B_m = \lfloor m/16 \rfloor$; and map also the integer n into a bin indexed by a bin number B_n computed by the function $B_n = \lfloor n/64 \rfloor$.

Step 4. Transform B_m into a 5-bit bin code $t = e_4e_3e_2e_1e_0$ for use as the authentication signal for P_c; and transform also B_n into a 3-bit bin code $s = h_2h_1h_0$ for use as the authentication signal for P_d.

3.2.2 Embedding of Authentication Signals

Because the color and depth image is saved as the PNG images, the alpha channels are used to hide the original bits for the process of original-image recovery. The original bits of a pair of a depth pixel and a color pixel includes the (2, 2, 1) LSBs of the original color-image pixel and the three LSBs of the original depth-image pixel. And the original bits of the depth and color image will be replaced with the authentication signals in this process. Therefore, in this process, the original bits of the depth and color image are hidden into the alpha channels of the depth and color image, respectively.

To embed the generated authentication signals generated as described above. At first we select a pixel P_d from the depth image I and a pixel P_c from the color image J both in a raster-scan order. Then, we consider pixel P_d and pixel P_c as the input to Algorithm 3.2.1 to generate the corresponding authentication signals, a 3-bit signal s for the color-image pixel P_c and a 5-bit signal t for depth-image pixel P_d. Besides, the original bits of P_d and P_c are hidden to the alpha value of the pixel P_c' and P_d', respectively. In order to achieve the goal that the pair of the depth and color images should be protected together, we embed the authentication signal *s*, which is generated from the pixel P_c of the color image into a pixel P_d' in the depth image. On the other hands, we embed the authentication signal *t*, which is generated from the pixel P_d of the depth image, into a pixel P_c' in the color image. In addition, both of the pixels, P_c' and P_d' are selected by using a secret key *K* and a random number generator *f*. When all the pixels in the color and the depth image have been processed, we get two protected images, a color and a depth image with authentication signals embedded.

The process described above for embedding the generated authentication signals is described in Algorithm 3.2.2 below.

Algorithm 3.2.2: embedding of authentication signals.

Input: a depth image *I*, a color image *J*, a random number generator *f*, and a secret key *K*.

Output: a depth image I_s with authentication signals embedded, and a color image J_s with authentication signals embedded.

Steps:

Step 1. In a raster-scan order, select a pixel P_d from the depth image *I* and a pixel P_c from the color image *J*.

Step 2. Perform the process of Algorithm 3.2.1 to generate two authentication signals, a 3-bit signal *s* for the depth pixel P_d and a 5-bit signal *t* for the color pixel P_c.

Step 3. Using the secret key *K* and the random number generator *f* to select randomly a pixel P_c' in the color image *J* and a pixel P_d' in the depth image *I*.

Step 4. Extract the original bits, the (2, 2, 1) LSBs of P_c, g_1, r_1, b_0, g_0, r_0, and the three LSBs of P_d, d_2, d_1, d_0; and embed g_1, r_1, b_0, g_0 into the four LSBs of the alpha-channel value of P_d' and embed r_0, d_2, d_1, d_0 into the four LSBs of the alpha-channel value of P_c'.

Step 5. Embed the 5-bit authentication signal $t = e_4e_3e_2e_1e_0$ of P_d into P_c' in J by replacing the (2, 2, 1) LSBs of the (R, G, B) values in the pixel P_c' with t; and embed the 3-bit authentication signal $s = h_2h_1h_0$ of P_c into P_d' in I by replacing the three LSBs of P_d' with s.

Step 6. If there remain unprocessed pixels in I or J, then go to Step 1; otherwise, take the processed I and J together with their alpha channels as the desired embedded images I_s and J_s in PNG format as output.

3.3 Authentication of KINECT Images

In this section, we introduce the proposed method for authentication of KINECT images. An illustration of the authentication signal extraction and verification process to apply the authentication function to the protected depth image is shown in Figure 3.2. In Section 3.3.1, the process of extracting the embedded authentication signals is described. Then, the process of detecting possible tampered regions in the depth or color image is described in Section 3.3.2.

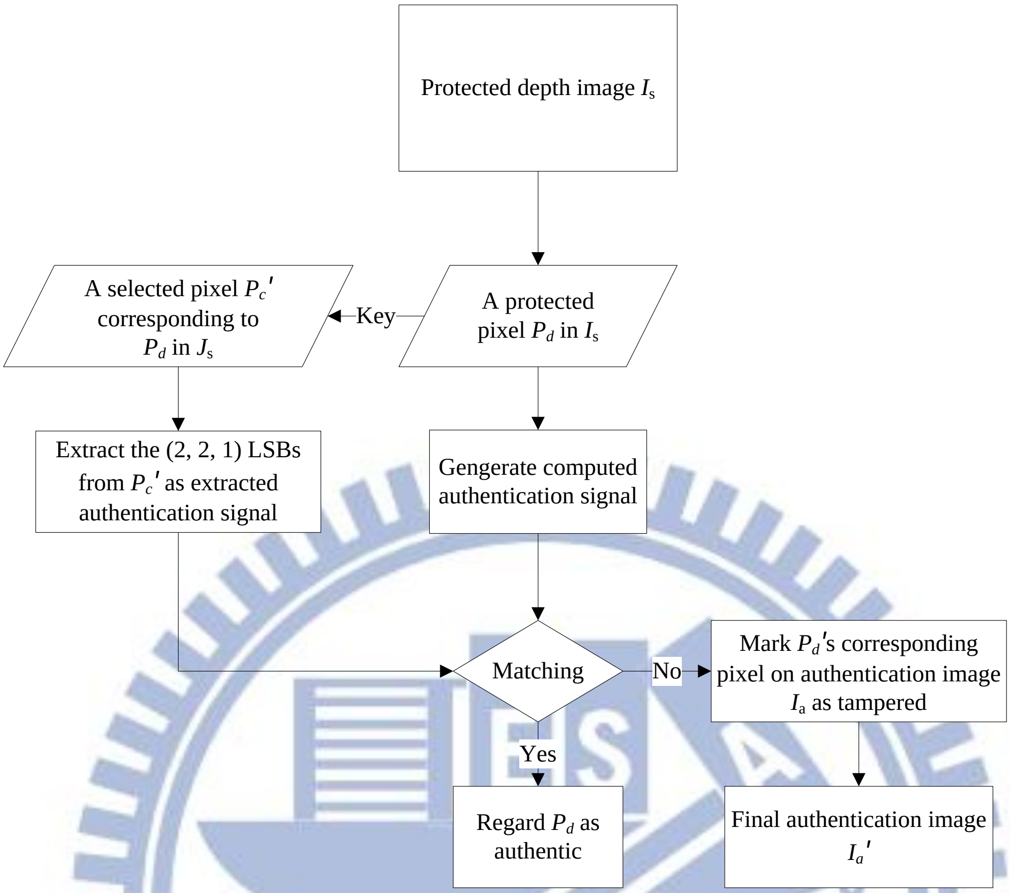

Protected depth image I_s
A selected pixel P_c' corresponding to P_d in J_s
Key
A protected pixel P_d in I_s
Extract the (2, 2, 1) LSBs from P_c' as extracted authentication signal
Gengerate computed authentication signal
Matching
No
Mark P_d's corresponding pixel on authentication image I_a as tampered
Yes
Regard P_d as authentic
Final authentication image I_a'

Figure 3.2 Illustration of proposed process of authentication signal verification and tampered pixel marking in protected depth images.

3.3.1 Extraction of Authentication Signals

In the process of authentication signal extraction, firstly we use the nine MSBs of each pixel P_d in the protected depth image to compute an authentication signal $t = e_4e_3e_2e_1e_0$, which is called the *computed authentication signal*. And according the random number generator f and the secret key K used in Algorithm 3.2.2, we find out the corresponding pixel P_c' in the protected color image. Then, we extract the (2, 2, 1) LSBs of the (R, G, B) values in the color pixel P_c' to form a binary string $t' = g_4'r_3'b_2'g_1'r_0'$. The string t' is called the *extracted authentication signal*. On the other hand, the other pair of the computed authentication signal s according to the color pixel

P_c' and the extracted authentication signal s' in the depth pixel P_d' also can be found out. Finally, the computed authentication signals s and t, and the extracted authentication signals s' and t' are used in the authentication signal verification process.

The detail of the process described above for the extractions of the embedded authentication signals and the computed authentication signals is presented as algorithm, Algorithm 3.3.1, below.

Algorithm 3.3.1: extraction of authentication signals.

Input: a color pixel value of P_c from a color image J_s; a depth pixel value P_d from a depth image I_s; the random number generator f and the secret key K used in Algorithm 3.2.2.

Output: the computed authentication signals s and t, and the extracted authentication signals s' and t'.

Steps:

Step 1. Transform the depth value of P_d into a 13-bit string d_{12}, d_{11}, d_{10}, …, d_0, and transform the three RGB color values of P_c into three eight-bit strings r_7, r_6, r_5, …, r_0, g_7, g_6, g_5, …, g_0, and b_7, b_6, b_5, …, b_0, respectively.

Step 2. Transform the nine MSBs, d_{12}, d_{11}, d_{10}, …, d_5, of d into an integer m, and concatenate the three 3-bit binary strings, r_7, r_6, r_5, g_7, g_6, g_5, b_7, b_6, b_5, in order into a string and transform the result into an integer n.

Step 3. Map the integer m into a bin indexed by a bin number B_m computed by the function $B_\text{m} = \lfloor m/16 \rfloor$; and map also the integer n into a bin indexed by a bin number B_n by the function $B_\text{n} = \lfloor n/64 \rfloor$.

Step 4. Transform B_m into a 5-bit bin code $t = e_4e_3e_2e_1e_0$ for use as the *computed authentication signal* from P_d; and transform B_n into a 3-bit bin code $s = h_2h_1h_0$ for use as the *computed authentication signal* from P_c.

Step 5. Use the random number generator f and the secret key K to select a color pixel P_c' corresponding to P_d from the color image I_s and a depth pixel P_d' corresponding to P_c from the depth image J_s.

Step 6. Transform the depth value of P_d' into a 13-bit string d_{12}' , d_{11}', d_{10}', …, d_0', and transform the RGB color values of P_c' into three 8-bit strings r_7', r_6', r_5', …, r_0', g_7', g_6', g_5', …, g_0', and b_7', b_6', b_5', …, b_0', respectively.

Step 7. Extract the three LSBs of P_d' to form a binary string $s' = d_2'd_1'd_0'$, and extract the (2, 2, 1) LSBs of the (R, G, B) values in pixel P_c' to form a binary string $t' = g_4'r_3'b_2'g_1'r_0'$, and call s' and t' the *extracted authentication signals*.

3.3.2 Detection of Tampering in Protected Depth and Color Images

During the process of detecting possibly tampered regions in the protected KINECT image, an authentication signal is computed from the nine MSBs of every depth pixel P_d. And the authentication signal, which is embedded in the color pixel P_c' corresponding to P_d selected randomly as described in Algorithm 3.2.2, is extracted. Then, the two authentication signals are compared with each other. If the two signals are different, the depth pixel P_d is regarded as having been tampered with. On the other hand, an authentication signal is computed from the (3, 3, 3) MSBs of every color pixel P_c. Also, the authentication signal embedded in the three LSBs of the corresponding depth pixel P_d' is retrieved. Then, we also compare the two authentication signals. If the signals are different, the color pixel P_c is regarded as having been tampered with. In addition, when mismatching occurs, the depth pixel P_d or the color pixel P_c is regarded as having been tampered with; and we mark its corresponding pixel in a corresponding blank image I_a or J_a as a black point to create two *authentication images* I_a' and J_a', respectively.

The detail of the process above for detection of tampering in the KINECT image is described in Algorithm 3.3.2 below.

Algorithm 3.3.2: tampering detection in protected depth and color images.

Input: a protected depth image I_s and a protected color image J_s, both with authentication signals embedded; two corresponding blank images, I_a and J_a; and the random number generator f and the secret key K used in Algorithm 3.2.2.

Output: an *authentication depth image* I_a' and an *authentication color image* J_a' corresponding to I_s and J_s, respectively.

Steps:

Step 1. Select a pixel P_d from I_s, and a pixel P_c from J_s, both in a raster-scan order, and perform the following steps.

Step 2. Take pixels P_d and P_c as inputs and apply Algorithm 3.3.1 to yield the *computed authentication signals, t* and *s*, and the *extracted authentication signals, s'* and *t'* for P_d and P_c, respectively.

Step 3. Match the computed authentication signal $t = e_4e_3e_2e_1e_0$ and the extracted authentication signal $t' = g_4'r_3'b_2'g_1'r_0'$ bit by bit; and if mismatching happens, regard P_d as having been tampered with and mark its corresponding pixel as a black point in the blank depth image I_a.

Step 4. Compare the computed authentication signal $s = h_2h_1h_0$ and the extracted authentication signal $s' = d_2'd_1'd_0'$; and if mismatching occurs, regard P_c as having been tampered with and mark its corresponding pixel as a black point in the blank color image J_a.

Step 5. If there exist unprocessed pixels in I_s or J_s, then go to Step 1; otherwise, take the final possibly marked I_a and J_a as authentication images I_a' and J_a', respectively.

3.4 Repairing and Recovery of Depth and Color Images

In this section, the proposed method of recovering and repairing the depth and color images is described. In Section 3.4.1, the process of repairing the tampered pixels is described. Then, the process of recovering the original images from images in which authentication signals are embedded is described in Section 3.4.2.

3.4.1 Repair of Tampering in Protected Depth and Color Images

As illustrated by Figure 3.3, at the beginning of the tampered-pixel repairing process, we can find the tampered pixels P_d' and P_c' respectively in I_s and J_s, which are the depth image and color image with authentication signals, according to the black points in the authentication images I_a and J_a, respectively. Then, the corresponding pixels P_c'' and P_d'' located at the positions where the authentication signals of P_d' and P_c' are embedded can be found out by the use of the random number generator f and the secret key K used in Algorithm 3.2.2. According to the extracted signals from P_c'' and P_d'', we can recover the bits which are used to generate these authentication signals. In addition, we pad an appropriate number of 0’s into the tails of these recovered bits as the repairing results.

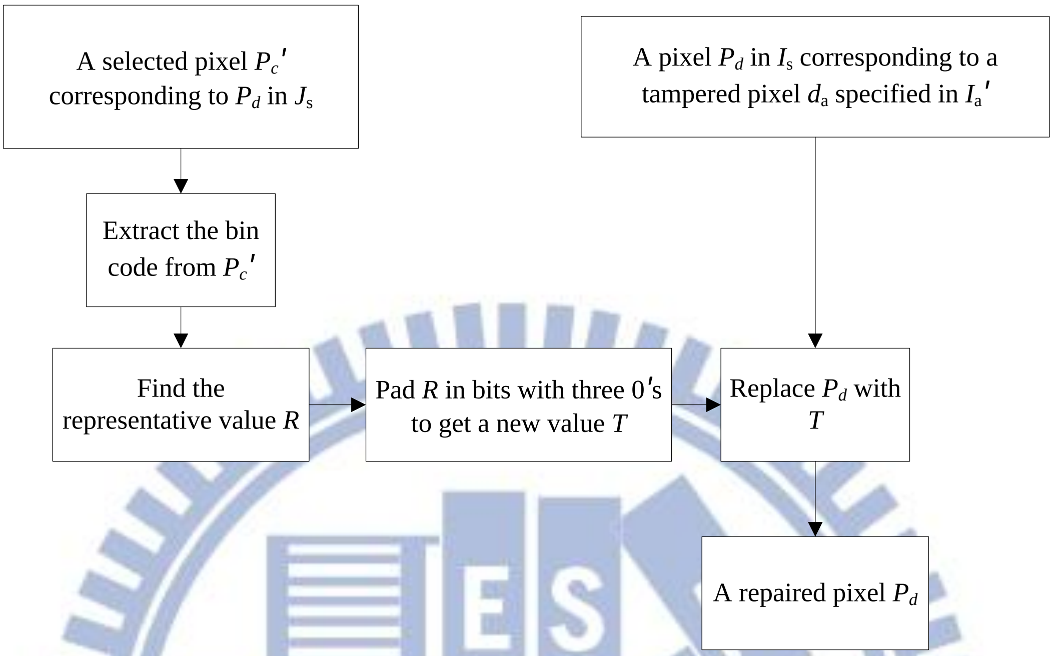

A selected pixel P_c' corresponding to P_d in J_s
Extract the bin code from P_c'
Find the representative value R
Pad R in bits with three 0′s to get a new value T
A pixel P_d in I_s corresponding to a tampered pixel d_a specified in I_a'
Replace P_d with T
A repaired pixel P_d

Figure 3.3 Illustration of the process of repairing tampered depth pixels.

The above process for repairing the tampered pixels in the color or depth image is described in more detail as an algorithm, Algorithm 3.4.1, as follows.

Algorithm 3.4.1: repairing the tampered pixels of the color and depth images.

Input: the depth image I_s and the color image J_s both with authentication signals embedded; corresponding authentication images I_a, and J_a; and the random number generator f and the secret key K used in Algorithm 3.2.2.

Output: a depth image I_r, and a color image J_r, which are repaired, if the input images have been tampered with.

Steps:

Step 1. Find each black point d_a in I_a' and each black point p_a in J_a', both in the raster-scan order, and find the pixel P_d' in I_s corresponding to d_a in position, and the pixel P_c' in J_s corresponding to p_a in position.

Step 2. Use the random number generator f with the secret key K to find the pixel P_c'' in J_s corresponding to P_d', where P_c'' is with a previously-embedded authentication signal; and use the random number generator f with the secret key K as well to find the pixel P_d'' in I_s corresponding to P_c' where P_d'' is with a previously-embedded authentication signal.

Step 3. Transform the RGB color values of P_c'' into three 8-bit strings r_7'', r_6'', r_5'', …, r_0'', g_7'', g_6'', g_5'', …, g_0'', and b_7'', b_6'', b_5'', …, b_0'', respectively; extract the (2, 2, 1) LSBs $g_1''r_1''b_0''g_0''r_0''$ from them; and transform $g_1''r_1''b_0''g_0''r_0''$ into an integer B_n' as the index of the bin; transform as well the depth value of d'' into a binary string d_{12}'', d_{11}'', d_{10}'', …, d_0'', extract the three LSBs d_2'', d_1'', d_0'' to form a binary string, and transform it into an integer B_m' as the index of the another bin.

Step 4. Repair the tampered depth pixel P_d' by the following steps.

4.1 Derive the representative value M_d of the bin indexed by B_m'.

4.2 Transform M_d into a 9-bit binary string $d_{12}d_{11}d_{10}d_9d_8d_7d_6d_5d_4$.

4.3 Pad three trailing 0's to $d_{12}d_{11}d_{10}d_9d_8d_7d_6d_5d_4$ to get a 13-bit string $T_d = d_{12}d_{11}d_{10}\ d_9d_8d_7d_6d_5d_4$0000.

4.4 Transform T_d into an integer u and replace the depth value of P_d' with u as the repairing result.

Step 5. Repair the tampered color pixel P_c' by the following steps.

5.1 Derive the representative value M_p of the bin indexed by B_n'.

5.2 Transform M_p into a 9-bit binary string $r_7r_6r_5g_7g_6g_5b_7b_6b_5$.

5.3 Separate the binary string into three parts, $r_7r_6r_5$, $g_7g_6g_5$, and $b_7b_6b_5$.

5.4 Pad five trailing 0's to each of the three binary strings to obtain respectively three new strings $T_r = r_7r_6r_5$00000, $T_g = g_7g_6g_5$00000, T_b

= $b_7b_6b_5$00000.

5.5 Transform these three 8-bit strings into three integers and replace the color values (R, G, B) of P_c' with them, respectively, as the repairing result.

Step 6. If there remain unprocessed black pixels in I_a or J_a, then go to Step 2; else, take the final I_s and J_s as the output repaired images I_r and J_r.

3.4.2 Recovery of Protected Depth and Color Images

At the beginning of the process of recovering the protected depth and color images, we can find the pixel P_d' in the depth image I_s with an authentication signal corresponding to P_c as well as the pixel P_c' in the color image J_s with an authentication signal corresponding to P_d by using the secret key K and the random number generator f. Then, the hidden bits in the alpha-channel values of P_d' and P_c' can be extracted and divided into two parts, one for the recovery of P_d and the other for the recovery of P_c. Furthermore, we can use these two parts to replace the (2, 2, 1) LSBs of P_c and the three LSBs of P_d, respectively. Finally, we can get the original color and depth images after all the pixels in J_s and I_s are processed.

The process described above for recovering the original bits in the depth and color images is described in more detail as an algorithm, Algorithm 3.4.2, as follows.

Algorithm 3.4.2: recovering the original color and depth images.

Input: a protected depth image I_s and a protected color image J_s both with authentication signals embedded; and the random number generator f and the secret key K used in Algorithm 3.2.2.

Output: the original depth and color images, I_o and J_o, respectively.

Steps:

Step 1. In a raster-scan order, select a pixel P_d from the protected depth image I_s and a pixel P_c from the protected color image J_s.

Step 2. Use the secret key K and the random number generator f to find the pixel P_d' corresponding to P_c in the protected color image J_s and the pixel P_c' corresponding to P_d in the protected depth image I_s.

Step 3. Extract the four LSBs, v_3, v_2, v_1, v_0, from the alpha-channel value of P_c'; and extract also the four LSBs, u_3, u_2, u_1, u_0, from the alpha-channel value of P_d'.

Step 4. Take u_3, u_2, u_1, u_0, v_3, in order as five original bits of P_c, concatenate them into a 5-bit string $T_c = u_3u_2u_1u_0v_3$, and take the other three bits, v_2, v_1, v_0, in order as three original bits of P_d.

Step 5. Replace the three LSBs of P_d with u_2, u_1, u_0 and transform the resulting 13-bit binary string of P_d into an integer as the original value of P_d; divide T_c into three parts, $S_\mathrm{r} = u_2v_3$, $S_\mathrm{g} = u_3u_0$, $S_\mathrm{b} = u_1$; replace the (2, 2, 1) LSBs of the (R, G, B) values of P_c with S_r, S_g, S_b, respectively; and transform the three resulting binary strings into three integers as the original R, G, B values of P_c.

Step 6. If there remain the unprocessed pixels in I_s or J_s, then go to Step 1; otherwise, take the final I_s and J_s as the original images I_o and J_o, respectively.

3.5 Experimental Results

The range of the depth values that are provided by the KINECT device is different from that of the general values of color-image pixels. In order to keep the complete depth information and display integrally the color and depth images as a whole, we transform firstly the value of each depth pixel into 13 binary bits and divide them into three parts, (5, 3, 5). Then, we take these parts as three binary strings and pad appropriate numbers of 0's to the tails of them, respectively. And the resulting three strings are transformed into the three integers. Finally, we take the three integers as the (R, G, B) values of the depth pixel, such that the depth information can be combined with the color information for displaying.

Some experimental results of applying the proposed method for authentication of KINECT images are shown in Figures 3.4 through 3.9. At the beginning of the experiment, we acquired color and depth images by a KINECT device. An example is shown in Figure 3.4. The corresponding protected color and depth images, into which the generated authentication signals are embedded by the proposed method, are shown in Figure 3.5. Then, we pretended to be a malicious user to tamper with the protected KINECT images, and the tampered images are shown in Figure 3.6. The tampered KINECT images were authenticated next by the proposed method, and the authentication results are shown in Figure 3.7. Furthermore, we took the authentication images and tampered images as inputs, and performed the process of repairing the tampered pixels. The results of the images which are repaired using the proposed method are shown in Figure 3.8. At last, we recovered the original KINECT images from the protected KINECT images in which authentication signals are embedded by using the proposed original-image recovering process. The results of the recovered images are shown in Figure 3.9.

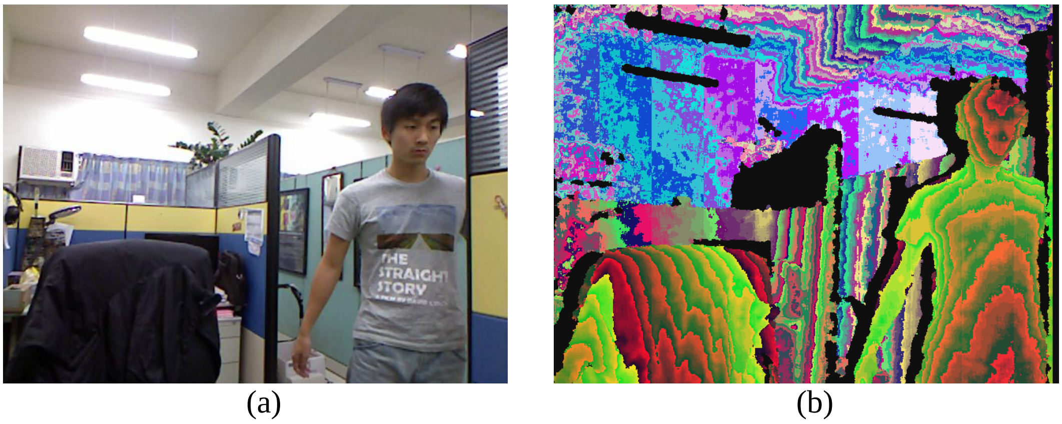

(a) (b)

Figure 3.4 The original KINECT images. (a) The color image. (b) The depth image.

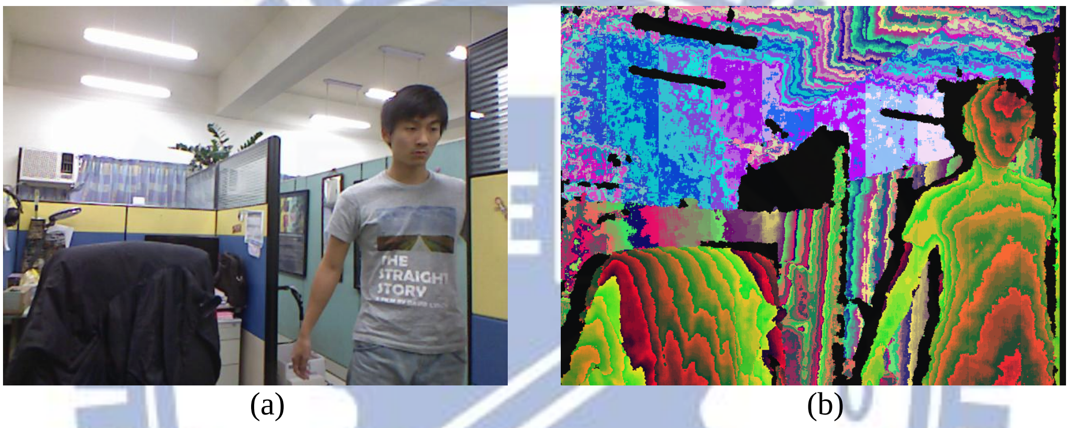

(a) (b)

Figure 3.5 The protected KINECT image with authentication signals embedded. (a) The color image with authentication signals embedded. (b) The depth image with authentication signals embedded.

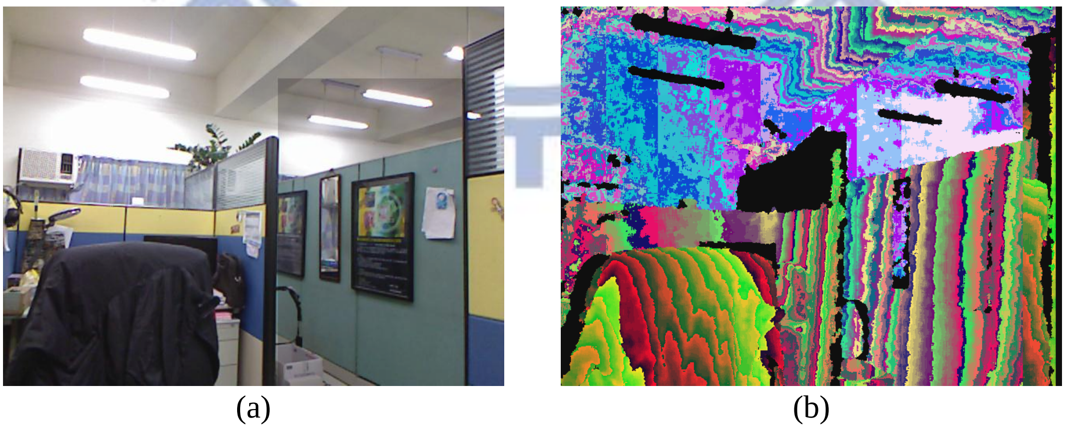
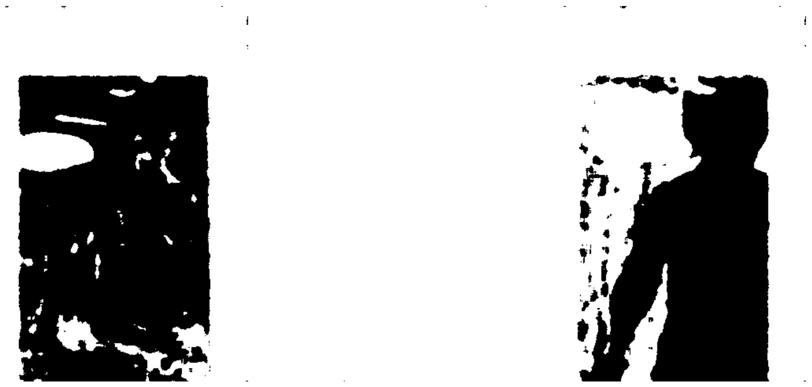
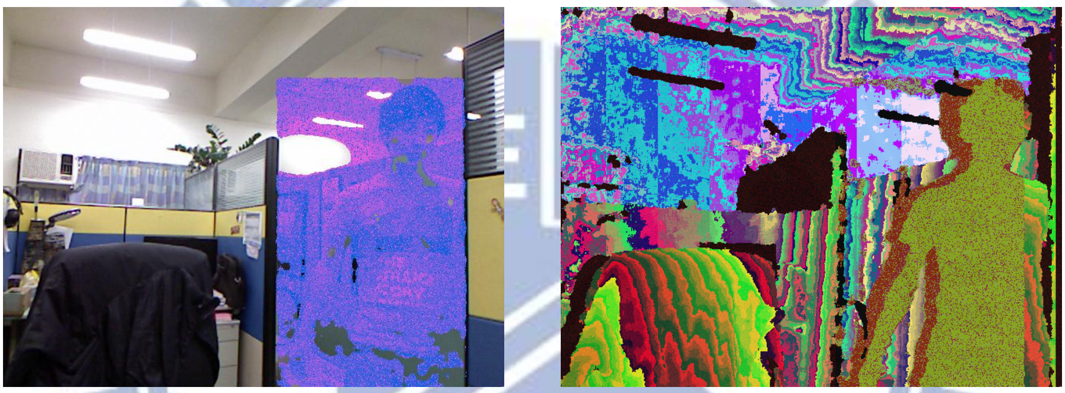

(a) (b)

Figure 3.6 The results of tampering protected KINECT images in which some regions have been modified. (a) The color result. (b) The depth result.



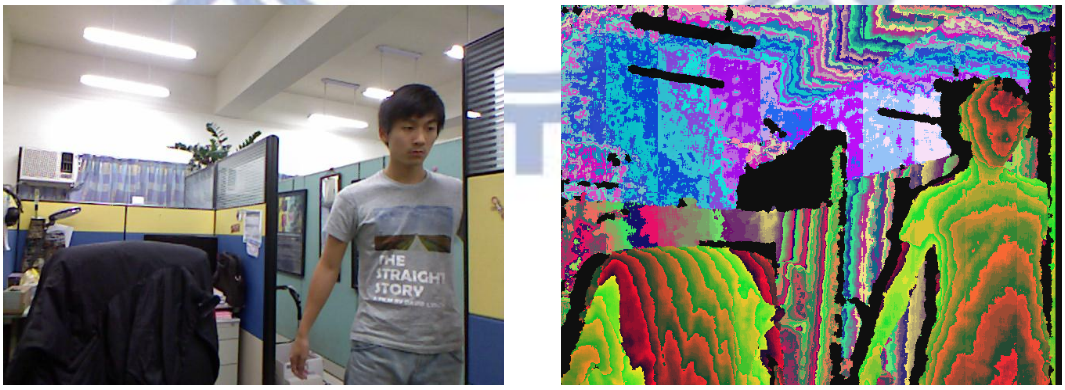
(a) (b)

Figure 3.7 The authentication result of the tampered KINECT images of Fig. 3.6. (a) The color authentication image. (b) The depth authentication image.



(a) (b)

Figure 3.8 The result of repairing the tampered KINECT images of Fig. 3.6. (a) The resulting color image. (b) The resulting depth image.



(a) (b)

Figure 3.9 The result of recovering the protected KINECT images in which authentication signals have been embedded. (a) The resulting color image. (b) The resulting depth image.

3.6 Discussions and Summary

In this chapter, a data hiding technique for authentication of KINECT images by embedding authentication signals into the depth and color images captured with the KINECT sensor is proposed. Based on the proposed method, we can generate the authentication signals according to the depth image and embed these signals into the color image. On the other hand, we also can generate authentication signals according to the content of the color image and embed these signals into the depth image. Then, the goal of protecting the color and depth image together is achieved.

In addition, to enhance the security of the proposed method, we select random positions in the depth or color images to embed authentication signals in order to reduce the possibility for a malicious user to figure out the locations of the embedded authentication signals in the KINECT images. Furthermore, if the protected KINECT images are tampered with, we can locate the pixels which are tampered with and repair some parts of the tampered regions by using the proposed method. The protected KINECT images with the authentication signals embedded can also be recovered by using the proposed recovering process.

Chapter 4

Covert Communication via KINECT Images by Interpolation at Depth Holes

4.1 Introduction

With the growing popularity of the Internet and KINECT devices, images acquired by KINECT devices are more commonly seen in daily life. Such images are shared frequently on the Internet. Because of these reasons, we propose in this study a method for covert communication via KINECT images by a new technique of interpolation at depth holes. The proposed method allows a user to hide secret messages imperceptibly in KINECT images. The user can so send the secret hiding result to persons he/she wants to communicate with. The detail of the proposed method is described in this chapter.

In the first part of this chapter, the definition of the problem and the idea of the proposed method are given in Sections 4.1.1 and 4.1.2, respectively. In Section 4.2, the proposed method for embedding secret messages into KINECT images by interpolation at depth holes is described. Next, the process of extracting the hidden secret messages is described in Section 4.3. Some experimental results of the proposed method are given in Section 4.4. Finally, a brief summary and some discussions are given in the last section of this chapter.

4.1.1 Problem Definition

Due to the hardware limitation of the KINECT device and possible object occlusions in the environment, pixels of interest in the depth image acquired by the KINECT device might be missed. The reasons of causing such missing values in the depth image might be: (1) that the distance between the KINECT device and the object is too short or too long; (2) that the surface of the object is too smooth to reflect the infrared light projected from the KINECT device; and (3) that the surface of the object is with high brightness or is infrared-absorbed. Objects like mirrors, glasses, fluorescent lamps, or other transparent substances might cause missing pixel values in the depth image. Pixels in the depth image with missing values are called *depth holes*.

Consequently, in this study we want to achieve the goal of enhancing the quality of the depth image while hiding secret messages in the KINECT images. The KINECT image, in which the secret message is embedded, cannot be distorted, or not too much if distortion is unavoidable. In addition, the changes caused by message embedding in the KINECT images cannot be too obvious. These requirements should be considered when designing a method for covert communication via KINECT images.

4.1.2 Proposed Idea

The idea of the proposed data hiding method for covert communication via KINECT images was mainly inspired by the features of KINECT images. The proposed method aims to enhance the quality of the depth image acquired by the KINECT device and hide the secret message into the KINECT image. The proposed method utilizes depth holes in the depth image and an interpolation technique proposed in this study to enhance the quality of the depth image while hiding the secret message in the KINECT images in the meantime.

Specifically, we transform the secret message into integers and generate the corresponding secret shares created by the use of the secret message by a secret sharing scheme [13]. Then, we search the depth holes in the depth image and compute accordingly the entire hiding capacity of the depth image. If the hiding capacity is enough for the current secret message, we use a secret key to randomize the order of data before embedding them into the depth holes by the proposed interpolation technique. In addition, we use the proposed interpolation technique as well to extract the hidden message in the depth image. The detailed algorithms about the proposed methods and the related processes of covert communication via KINECT images are presented in the following sections.

4.2 Interpolation at Depth Holes for Embedding of Secret Messages

In this section, we will introduce the details of the implemented process of interpolation at depth holes for embedding the secret message. An illustration of the process is illustrated in Figure 4.1. The detailed process of transforming the secret message is described in Section 4.2.1. Then, the detailed process of embedding the secret message by interpolation at depth holes in described in Section 4.2.2.

4.2.1 Transformations of Secret Messages

At the beginning of the proposed process, we transform a given secret message into decimal integers. Then, we transform these decimal integers into a binary string *S* and use the secret key *K* to randomize the string *S*. And we transform the string *S* into a string of integers S_i. Then, we generate secret shares using the string S_i by the secret sharing scheme by Shamir [13]. In addition, we transform the generated secret shares into a binary string S_r as the output of the hiding data.

As a review of the secret sharing scheme proposed by Shamir [13], a secret message M in the form of integers is transformed into shares. Then, these shares are distributed to the secret-sharing participants to keep. And as long as k of the n shares are collected, the original secret message M can be recovered, where $k \leq n$.

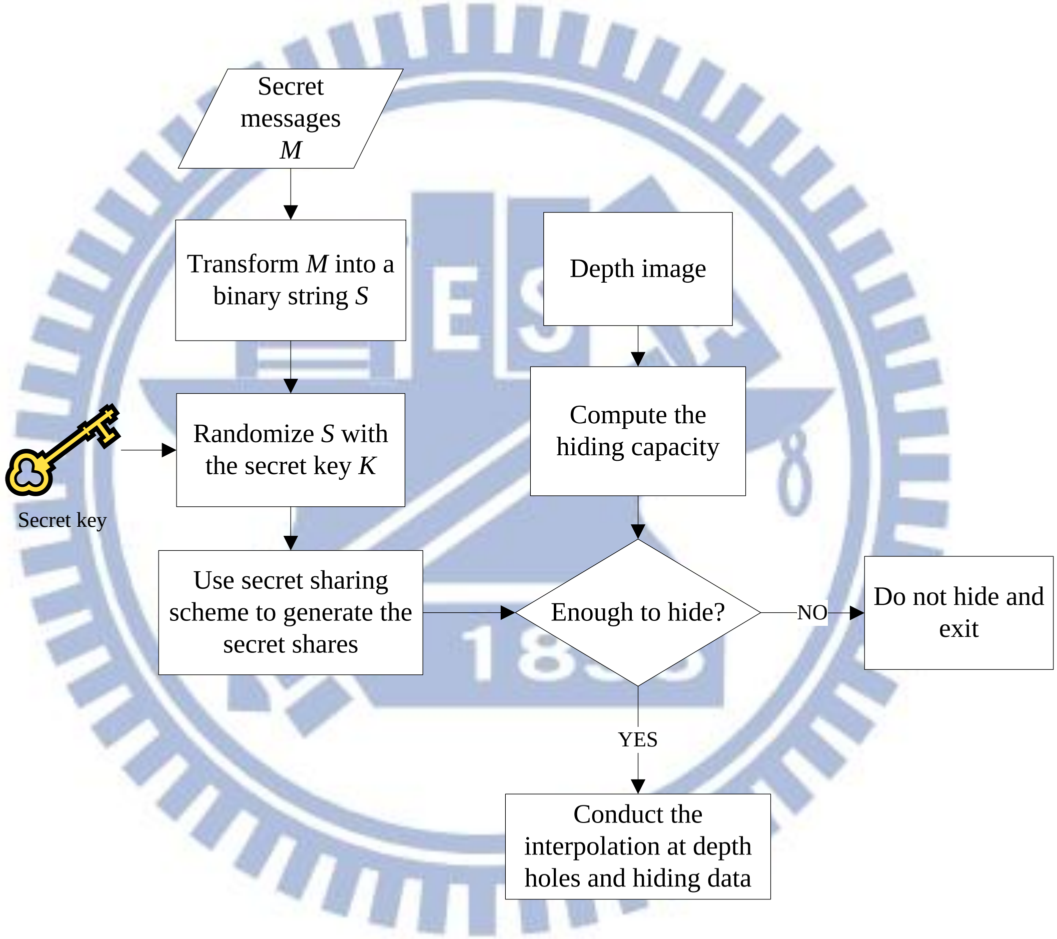

Secret messages M
Transform M into a binary string S
Depth image
Randomize S with the secret key K
Compute the hiding capacity
Secret key
Use secret sharing scheme to generate the secret shares
Enough to hide?
NO
Do not hide and exit
YES
Conduct the interpolation at depth holes and hiding data

Figure 4.1 Illustration of secret-message transformation and hiding in the depth image.

Algorithm 4.2.2: processing of the secret messages.

Input: a secret message M and a secret key K.

Output: a data string S for hiding with its bits in a randomized order.

Steps:

Step 1. Transform the secret message M into integers character by character according to the ASCII codes.

Step 2. Decrease the values of these integers by subtracting 96 from each of them.

Step 3. Transform the resulting integers into a binary string S.

Step 4. Randomize the order of the bits in S by using the secret key K.

Step 5. Transform string S into a string of integers S_i.

Step 6. Generate n secret shares according to the string S_i by the Shamir secret sharing scheme [13].

Step 7. Transform the resulting secret shares into a binary string S_r as the output hiding data string.

4.2.2 Interpolation at Depth Holes

In the proposed process of interpolation at depth holes, at first we use a 3×3 block window to search all the depth holes in the depth image. If there contain two depth values around the center of a 3×3 block, then we consider the center pixel of the block to be a depth hole, i.e., a depth pixel whose depth information is missing. And we compute the hiding capacity H_c and compare H_c with the length of the data string S. If the H_c is larger than the length of S, we use the secret key K to decide which depth hole should hide the secret data string S.

In the proposed interpolation scheme, we use the parameters around the center of the block to decide the value of a depth hole whose depth value is missing. First, we compute the distances d_i between these parameters C_i and the depth hole H in the block. In addition, a *weighted number* W is decided by every two bits in order of a data string S

for hiding and a code table, Table 1. Finally, the value of the depth hole is computed by in terms of the distances d_i of the parameters C_i and the weighted number W.

The proposed technique for interpolation at depth holes as described above is presented as an algorithm, Algorithm 4.2.1, as follows.

Table 4.1 Two bits corresponding to the different weights.

Two bits to be hidden	Corresponding weight
00	-0.25
01	-0.5
10	0.25
11	0.5

Algorithm 4.2.1: interpolation at depth holes.

Input: a cover depth image I acquired by the KINECT device, the data string S of a secret message M for hiding, a secret key K and a random number generator f.

Output: a stego-depth image I_s with the secret message M hidden in it.

Steps:

Step 1. In a raster-scan order, use a 3×3 block B as a "moving window" to search all the depth holes in the depth image I and compute the hiding capacity H_c as the total number of all found depth holes.

Step 2. Compare H_c with the length L_s of string S: if $L_s \leq H_c$, then go to Step 3; otherwise, do not hide the string S and exit.

Step 3. Set the index x of the depth hole to 0;

Step 4. Find a depth hole H in a raster-scan order.

Step 5. According to the non-zero values, called parameters C_i, around the depth hole *P* in the 3×3 block *B*, decide the corresponding distance d_i of C_i. If C_i is the four neighbors of *P*, the corresponding distance d_i is 1; otherwise, the corresponding distance d_i is $\sqrt{2}$.

Step 6. According to *x*, H_c and the secret key *K*, decide whether to hide data at this depth hole *P* or not by the following steps.

6.1. Use H_c, the secret key *K*, and the random number generator *f* to generate a random string *R* with values ranging from 0 to H_c and length equal to L_s.

6.2. If the index value *x* of the depth hole *H* is in the random string *R*, then go to the Step 7 (to perform data hiding); otherwise, set the *weighted number W* to 0 and go to Step 9.

Step 7. Take two bits b_0b_1 in order from the data string *S* of the secret message *M* for hiding.

Step 8. According to b_0b_1 and Table 1, decide a *weighted number W*.

Step 9. Compute the value of the depth hole *H* by

$$H = \frac{\sum_{i=1}^{n}\left(\frac{1}{d_i}\right)^2 C_i}{\left(\sum_{i=1}^{n}\left(\frac{1}{d_i}\right)^2\right) + W},$$

where $n \leq 8$. And update the index *x* by $x = x + 1$

Step 10. If there remain unprocessed bits in *S*, then go to Step 4; otherwise, take the processed *I* as the desired depth image I_s with the secret message *M* hidden.

4.3 Extraction of Secret Messages

In this section, we will introduce the details of the implemented process of extracting the secret message from the depth image. The detailed extraction process of the hidden data in the depth image is described in Section 4.3.1. Then, the detailed process of transforming the hidden data into the original secret message is described in Section 4.3.2.

4.3.1 Extraction of embedded data from depth image

In the proposed process of extracting the secret message, at first we use a 3×3 block window to search the pixel which is embedded in the depth image. When the value of the center pixel in the block is non-zero, we can decide whether message data are embedded in the center pixel or not by comparing the value of the center pixel in the block with *the computed values* of the block, where the computed values of the block include four values which are computed according to the parameters around the center pixel in the block, the corresponding distances, and Table 1. In addition, we can extract the two message bits which are hidden in the center pixel if one of the computed values equals value of center pixel. Finally, we can search in the depth image block by block, and in this way a binary string *S* can be extracted from the depth image.

The proposed process above for message extraction from the depth image is described as an algorithm, Algorithm 4.3.1, as follows.

Algorithm 4.3.1: extraction of the embedded data from the depth image

Input: a depth image I_m with a hidden secret message.

Output: a string S_e representing the hidden message.

Steps:

Step 1. In a raster-scan order and using a 3×3 block B as a "moving window," search the input I_m block by block, and if the value V_c of the center pixel P in the block B is non-zero, then decide that message bits have been embedded into a block B and go to Step 2; otherwise, go to Step 1 to search the next block.

Step 2. According to the non-zero values, called parameters C_i, around the center pixel P in the 3×3 block B, decide the corresponding distance d_i of C_i and according to Table 1, decide the value of weight W_k, where k = 0, 1, 2, 3. If C_i is the four neighbors of P, the corresponding distance d_i is 1; otherwise, the corresponding distance d_i is $\sqrt{2}$.

Step 3. Compute *the computed values* R_0, R_1, R_2, R_3 by

$$R_k = \frac{\sum_{i=1}^{n}\left(\frac{1}{d_i}\right)^2 C_i}{\left(\sum_{i=1}^{n}\left(\frac{1}{d_i}\right)^2\right) + W_k},$$

where k = 0, 1, 2, 3.

Step 4. Compare the value V_c with the four computed values R_0, R_1, R_2, R_3: if one of the computed values equals V_c, then decide that message bits have been embedded into block B, extract from B two message bits according to the computed value and Table 1, and append these two bits to the end of string S_e.

Step 5. If there remain unprocessed block in I_m, then go to Step 1; otherwise, take the final string S_e as the desired extracted string S_e.

4.3.2 Transformation of the extracted data

At the beginning of the proposed process of transforming the extracted string S_e into the original secret message, we transform the extracted string S_e into a string of integers which are the *n* secret shares. Then, we recover the string S_i of integers from the *n* secret shares by an inverse version of the Shamir secret sharing scheme [13]. In addition, we transform the string S_i into a binary string *S* and use the secret key *K* to resume the order of the bits in the string *S*. Next, we transform the string *S* into the original string of integers and increase the values of these integers by adding 96 to each of them. Finally, the original secret message can be acquired by transforming these integers into the corresponding characters.

The proposed process above for transforming the extracted data into the original secret message is described as an algorithm, Algorithm 4.3.2, as follows.

Algorithm 4.3.2: transform the extracted data.

Input: an extracted string S_e and a secret key *K*.

Output: the secret message *M*.

Steps:

Step 1. Transform the extracted string S_e into a string of integers which are the *n* secret shares.

Step 2. Recover the string S_i of integers from the *n* secret shares by an inverse version of the Shamir secret sharing scheme [13].

Step 3. Transform the string S_i into a binary string *S*.

Step 4. Use the secret key *K* to resume the order of the bits in string *S*.

Step 5. Transform the string *S* into the original string of integers.

Step 6. Increase the values of these integers by adding 96 to each of them.

Step 7. Recover the secret message *M* by transforming these integers one by one according to the ASCII codes and concatenate them in order.

4.4 Experimental Results

Because of the reasons which were mentioned in Chapter 3, we also transform the value of each depth pixel into 13 binary bits and divide them into three parts, (5, 3, 5). Next, we take these parts as three binary strings and pad appropriate numbers of 0's to the tails of them, respectively. Then, we transform the resulting three strings into the three integers. Finally, we take the three integers as the (R, G, B) values of the depth pixel, such that the depth information can be combined with the color information for displaying.

Some experimental results of applying the above proposed method for covert communication via KINECT images by interpolation at depth holes are shown in Figures 4.2 through 4.5. At the beginning of the experiment, the original depth image acquired by a KINECT device is shown in Figure 4.2(a). In addition, we transform the secret message *M* into a binary string *S* for hiding by Algorithm 4.2.2. If the hiding capacity is large enough to hide the string *S*, then we use the secret key *K* to randomize the order of *S* and hide the resulting string into the depth image by the proposed method of interpolation at depth holes. Figure 4.2(b) is the result with Figure 4.2(a) as input. Two more results are shown in Figures 4.3 and 4.4. Finally, we used the proposed extraction process to extract the hidden data and recover the hidden data to compose the secret message. Two results are shown in Figure 4.5 where Figure 4.5(a) was the result extracted with a right key and Figure 4.4(b) shows the result extracted with a wrong key.

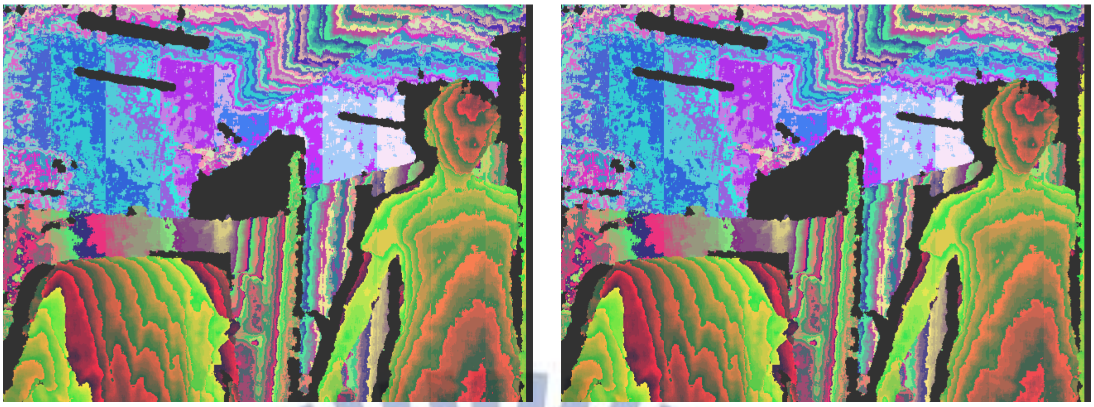


Figure 4.2 The original depth image acquired by KINECT devices (left) and the depth image with the secret message embedded (right).

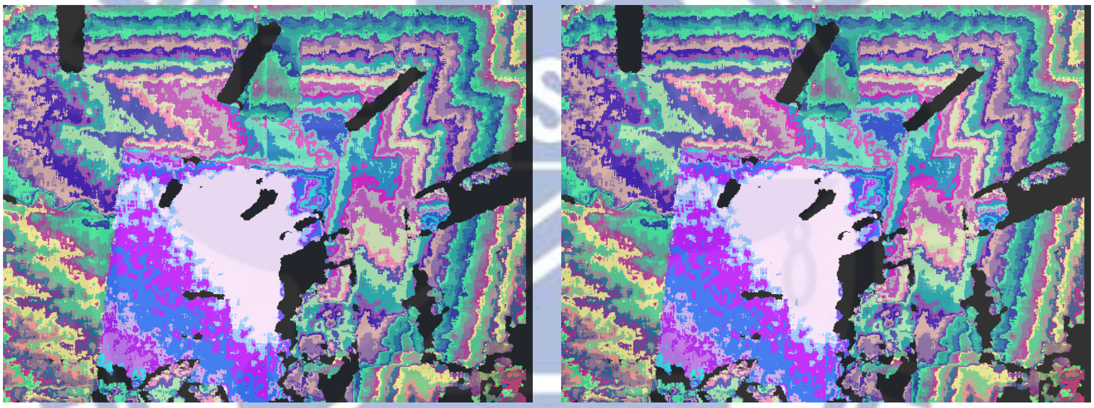


Figure 4.3 The original depth image of CVLAB (left) and the depth image with secret message embedded (right).

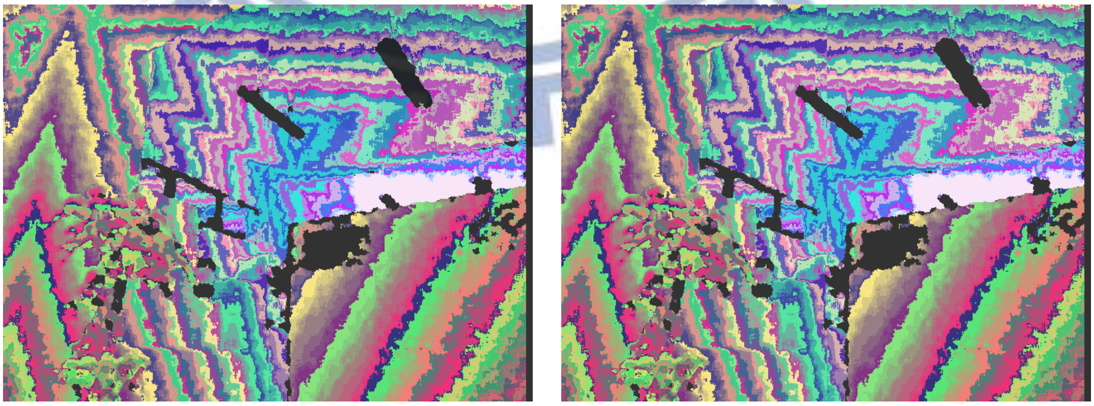
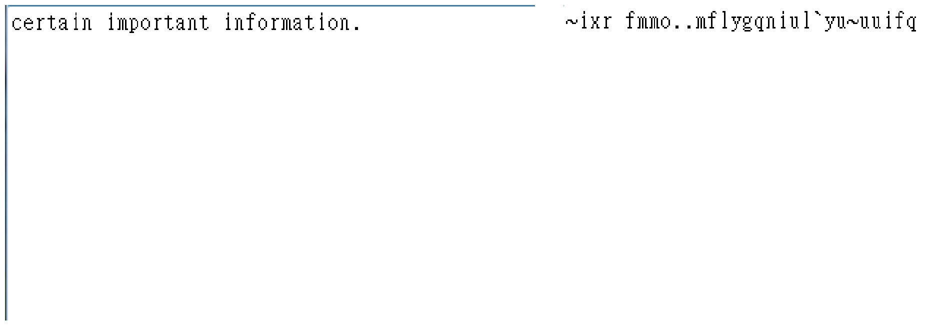


Figure 4.4 The original depth image of the other view in CVLAB (left) and the depth image with secret message embedded (right).



```
certain important information.
```

```
~ixr fmmo..mflygqniul`yu~uuifq
```

(a) (b)

Figure 4.5 Extracted secret messages. (a) The correct secret message extracted with a right key. (b) The incorrect secret message extracted with an erroneous key.

4.5 Discussions and Summary

In this study, a data hiding method for covert communication via KINECT images by interpolation at depth holes is proposed. The proposed method hides the secret message into the depth image, which is acquired by the KINECT device, by using the feature, depth holes, of depth images. To achieve the aims of enhancing the quality and hiding the secret message, the proposed method utilizes the depth holes in the depth image and a new interpolation technique to enhance the quality of the depth image and hide the secret messages in the KINECT image in the meantime. The secret sharing scheme [13] and a secret key are also used in the proposed method to transform the secret message for the reasons of enhancing the security and robustness of the secret message.

Some discussions about security issues of the above proposed method are as follows. In the transformation process, the secret message is transformed into a binary string S, first. Then, we use the secret key to randomize the order of the string S and transform string S into a string of integers S_i. In addition, according to the string S_i we can generate the n secret shares by the secret sharing scheme. Then, transform these n

secret shares into a binary string as the data string to hide in the depth image by the proposed interpolation method. Therefore, even if the malicious user can extract the string from the interpolated depth image, it still can’t be transformed back to the secret message without the secret key. And if one of the secret shares is modified by the malicious user, the original secret message still can be recovered by using the other secret shares and the inverse version of the Shamir secret sharing scheme [13].

Chapter 5

Copyright Protection of KINECT Images by 3D Visible Watermarking

5.1 Introduction

Because of the ubiquity of KINECT devices, people can use a KINECT device to acquire the color and depth information of some unique objects. Besides, information acquisition and sharing are becoming more and more convenient on the Internet nowadays. Consequently, the copyright protection of the color and depth information of highly-concerned objects has become an important issue. Therefore, a method for copyright protection of KINECT images by 3D visible watermarking is proposed in this study, and the detail of this method is described in this chapter.

In Sections 5.1.1 and 5.1.2, the related problem definitions and the idea of the proposed method are given. In Section 5.2, a method for transforming a pair of KINECT images, namely, a color image and a depth image, into a 3D image is described. The proposed process for embedding and eliminating of the 3D visible watermark in KINECT images by *difference expansion* is presented in Section 5.3. Then, another process proposed for embedding and eliminating of the 3D visible watermark in KINECT images by *revisable contrast mapping* is described in Section 5.4. In Section 5.5, some experimental results are presented. In the last section of this chapter, some discussions and a summary are given.

5.1.1 Problem Definition

As the popularity of KINECT device increases, related applications are also growing in many fields. A depth image acquired by the KINECT device contains more useful information than a commonly-seen 2D image. For example, we can use the KINECT device to capture the depth image of some architecture, and then transform the depth image into the 3D coordinates of the architecture for various digital applications. Therefore, the acquisition of the 3D coordinates of objects is more easily than before. However, the copyright of these kinds of information will also become more important.

For the reasons above, in this study we want to add a 3D visible watermark into each of the KINECT images, namely, the depth image and the color image, to protect the copyright of such information. In addition, embedding the 3D visible watermark into a KINECT image might cover some region of the original KINECT image. So the original information of the covered region should be embedded into the watermarked KINECT image, so that we can extract later this information from the watermarked KINECT image to recover the covered region. And the recovery information can be acquired only by an authorized person who has a secret key. That is, only a person who has the secret key can remove the 3D watermark from the watermarking KINECT image and simultaneously recover the covered region with the recovery information. These conditions above should be considered when designing a method for copyright protection of KINECT images, as have been done in this study.

5.1.2 Proposed Idea

The basic idea of the proposed method for copyright protection in KINECT images is to embed a 3D *visible* watermark into the KINECT image to represent the ownership of the KINECT image. In the proposed method, first we combine the color image and the depth image together to form a *3D image*. And according to the 3D coordinates in the image, we can find the corresponding color information in the color image. Therefore, we can process 3D image data which are formatted as (x, y, z, r, g, b), including the information of the depth and color images. And then we embed a 3D visible watermark into the 3D image data for protecting the copyright. After this process, some regions of the original 3D image are replaced by the information of the 3D visible watermark. We hide such regions in the 3D image by the difference expansion method proposed by Tian [6] and by the reversible contrast mapping method proposed by Coltuc and Chassery [8], respectively. In addition, only authorized users who have the secret key can extract the recovery information from the watermarked 3D image. And according to the recovery information, we can remove the 3D watermark from the 3D image and recover the original contents of the covered regions *losslessly*.

Detailed algorithms describing the proposed methods and the related processes of copyright protection of KINECT images are presented in the following sections.

5.2 Transformation of KINECT images into 3D images

In this section, the details of the transformation of KINECT images into 3D images are described. For the reason of displaying the features of KINECT images, we transform a depth image and a color image into a 3D image with 3D space coordinates. Because the data in the depth image, which are acquired by KINECT devices, are the depth values with 2D coordinates, we want to transform the 2D coordinates in the depth image into the 3D space coordinates. For this purpose, we utilize the principle of the pinhole camera model.

An illustration of the principle of the pinhole camera model is shown in Figure 5.1. Our purpose is to transform a point *I* with the 2D coordinates (*u*, *v*) in the depth image into a point *G* with 3D space coordinates (*X*, *Y*, *Z*) in the camera coordinate system, as illustrated in Figure 5.1. In the figure, the value *D* is a pixel value in the depth image, *f* is the focal length of the IR camera in the KINECT device, and (*u*, *v*) are the 2D image coordinates. Then, according to the Pythagorean Theorem, the value of *d* can be computed by the following equation:

$$d = \sqrt{u^2 + v^2 + f^2}.$$

Next, according to the similar-triangle principle, we can derive the relationship of *X*, *Y*, *Z*, *D*, *u*, *v*, *f*, *d* to be:

$$\frac{X}{u} = \frac{Y}{v} = \frac{Z}{f} = \frac{D}{d}.$$

According to these relationship and the value $d = \sqrt{u^2 + v^2 + f^2}$, we can derive the following equations which transform a point *I* with 2D coordinates (*u*, *v*) into a point *G* with 3D space coordinates (*X*, *Y*, *Z*) by:

$$X = u \cdot \frac{D}{\sqrt{u^2 + v^2 + f^2}}, \quad Y = v \cdot \frac{D}{\sqrt{u^2 + v^2 + f^2}}, \quad Z = f \cdot \frac{D}{\sqrt{u^2 + v^2 + f^2}}.$$

Finally, we can find the corresponding color information of each 3D space point in the 3D image by the above process. Consequently, a pair of KINECT images, a color image and a depth image, can be transformed into a 3D image whose pixel value may be formatted as (*x*, *y*, *z*, *r*, *g*, *b*).

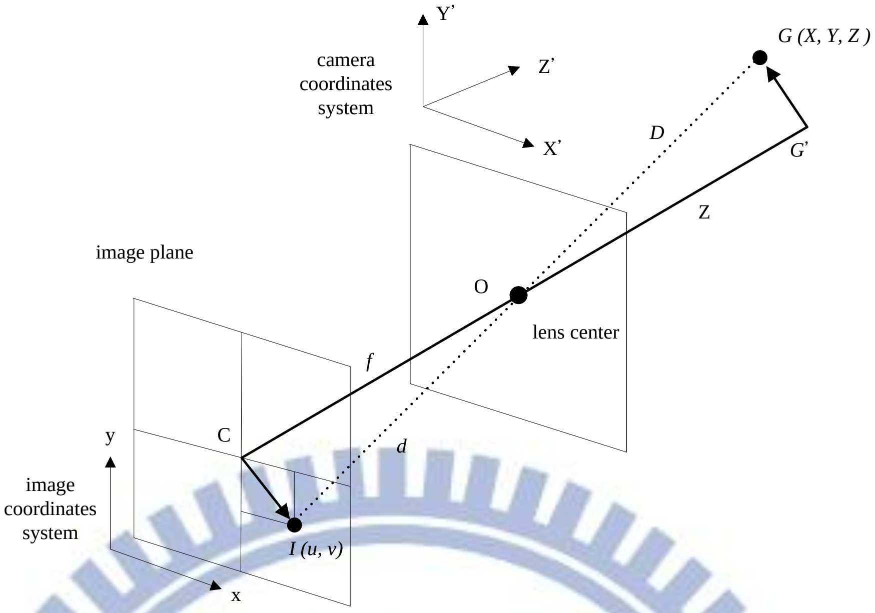


Figure 5.1 Illustration of the pinhole camera model.

5.3 Embedding and Eliminating 3D Visible Watermarks in KINECT Images by Difference Expansion

In this section, the details of the proposed method for embedding and eliminating a 3D visible watermark in each KINECT image by the difference expansion method are described. An illustration of the process is shown in Figure 5.2. The detailed process of embedding the 3D visible watermark into a KINECT image and hiding the recovery information in the watermarked 3D image is described in Section 5.3.1. In Section 5.3.2, the process of extracting the recovery information from the water marked 3D image is described. An illustration of the process of eliminating the 3D visible watermark is shown in Figure 5.3. And in Section 5.3.3, the process of eliminating the 3D visible watermark and recovering the covered region is described. Some experimental results of the process of embedding and eliminating 3D visible watermarks in KINECT image by difference expansion are given in Section 5.3.4.

5.3.1 Embedding of 3D Visible Watermarks

In the process of embedding of a 3D visible watermark into a KINECT image (a color image or a depth image), we first transform every pixel value in the depth image into the corresponding 3D real-world coordinates (x, y, z). Then, according to these coordinates, we try to find the corresponding color information (r, g, b) for each pixel. In this way, the original depth and color images are combined together into a 3D image I, which includes all the information of the KINECT images, with its coordinates formatted as (x, y, z, r, g, b).

In addition, the information of the 3D visible watermark is also formatted as (x, y, z, r, g, b), such that we can embed the 3D visible watermark W into the 3D image I. Then, the 3D visible watermark W will cover some region of the original 3D image I. And we transform the data, which are covered by the 3D visible watermark W, into a recovery-data string S and use a secret key K to randomize the order of the bits in the string S. Finally, we hide the string S into the watermarked 3D image I' by the difference expansion method proposed by Tian [6].

In more detail, during the data hiding process, at first we partition the watermarked 3D image I' into pairs. A pair consists of two neighboring points and the pairing is done horizontally. Then, in a raster-scan order, we index each pair in the watermarked 3D image I'. Next, we use a secret key K and a random number generator f_r to generate a random number sequence H. According to the order of the random sequence H, we can use an index i of H to find a pair of two neighboring points $V_1(x_1, y_1, z_1, r_1, g_1, b_1)$ and $V_2(x_2, y_2, z_2, r_2, g_2, b_2)$ with V_1 being with the index i. In addition, if the difference of V_1 and V_2 is small enough, then we hide the bits taken from the string S in the difference of the pair of by expanding the difference. For example, we hide one bit b in the difference d_r of r_1 and r_2 by taking two times the value of d_r and plus b as the hiding result, which is called the *expanded difference dr'*. In addition, according to *dr'*, we change the values of r_1 and r_2 into two new ones r_1' and r_2' for the purpose of enforcing the expanded difference *dr'* to keep the average value unchanged.

For instant, if the pair (r_1, r_2), $r_1 = 6$, $r_2 = 5$ and the difference d_r is 1. And if we want to hide the bit *b* which is “1” into the pair, then we expand the difference *dr* to be the *expanded difference dr'* = $2 \times d_r + b = 3$. And since the *floor* of the original average value *l* between $r_1$and r_2 is 5, we change r_1 to $r_1' = l + \lfloor (d_r' + 1) / 2 \rfloor = 7$ and change r_2 to $r_2' = l - \lfloor (d_r') / 2 \rfloor = 4$, which keep the average to be $\lfloor (7 + 4) / 2 \rfloor = 5$. And finally, we take the pair (r_1', r_2') as the final result to be embedded. In this way, we can hide the string *S* into the watermarked 3D image *I'*.

The detail of the above proposed process for embedding the 3D visible watermark is presented as an algorithm, Algorithm 5.3.1, below.

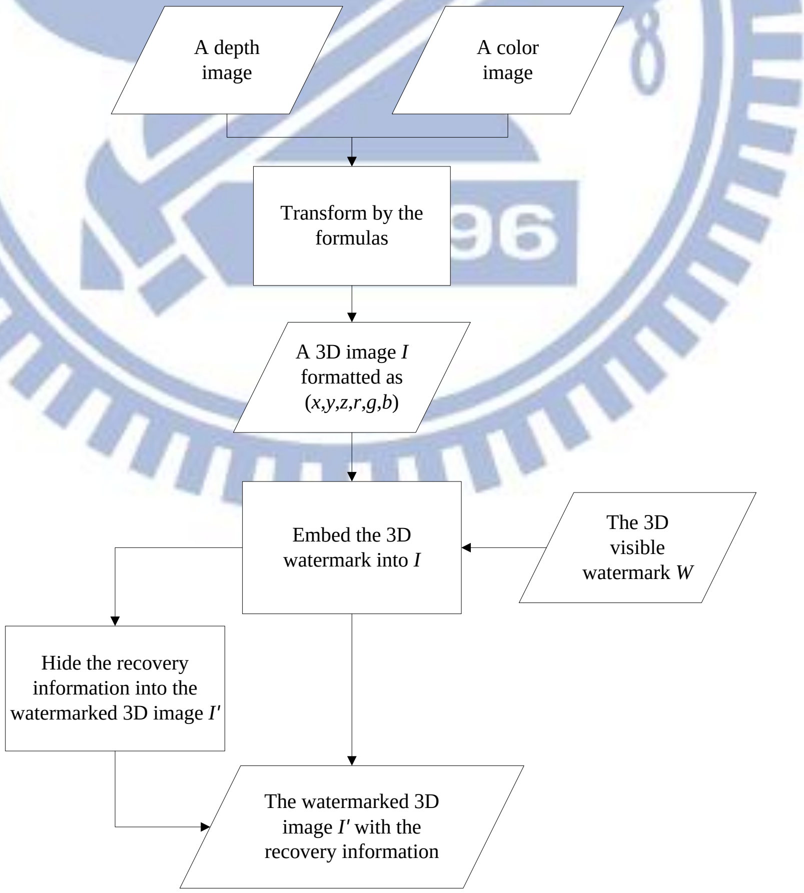

A depth image
A color image
Transform by the formulas
A 3D image *I* formatted as (*x*,*y*,*z*,*r*,*g*,*b*)
Embed the 3D watermark into *I*
The 3D visible watermark *W*
Hide the recovery information into the watermarked 3D image *I'*
The watermarked 3D image *I'* with the recovery information

Figure 5.2 Illustration of image transformation and 3D watermark embedding.

Algorithm 5.3.1: transformation and embedding of the 3D watermark.

Input: a pair of KINECT images, which includes a depth image and a color image, a secret key K, a random number generator f_r, and a 3D visible watermark W.

Output: a watermarked 3D image I' with the recovery information embedded.

Steps:

Step 1. Transform the depth image and the color image into a 3D image I which is *formatted* as (x, y, z, r, g, b) by the transformation process described in Section 5.2.

Step 2. According to the positions described by $(x_w, y_w, z_w, r_w, g_{\mathrm{w}}, b_w)$ of all the pixels of W, deicide a region R in I whose pixels' coordinates are computed by the following formulas which are derived in a way similar to that discussed in Section 5.2:

$$u_c = x_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}}, \quad v_c = y_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}},$$

where (x_w, y_w, z_w) are the pixels' coordinates of the 3D watermark W and (u_c, v_c) are the pixels' coordinates of the region R in the 3D image I covered by the 3D watermark W.

Step 3. Transform the data (x, y, z, r, g, b) of the position (u_r, v_r) in the covered region R into (d, r, g, b) by replacing (x, y, z) with the depth value d of the depth pixel (u_r, v_r) in the depth image for reducing the data that need to be hidden; and then transform every (d, r, g, b) into a recovery binary string S by the following steps: (1) transform each d into a 13-bit string S_1; (2) transform each of r, g, and b into three 8-bit strings S_2, S_3, and S_4; (3) combine S_1 through S_4 into a string T; and (4) append T to string S.

Step 4. Embed the 3D watermark W into the 3D image I by replacing the data of the region R in I with the data of W.

Step 5. Use the secret key K and the random number generator f_r to randomize the order of the bits in string S, and use K and f_r again to generate a random-number sequence H.

Step 6. Hide the string S into the watermarked 3D image I' by the difference expansion method as follows.

6.1. According to the order of the random numbers in H, find a pair of two neighboring points $V_1(x_1, y_1, z_1, r_1, g_1, b_1)$ and $V_2(x_2, y_2, z_2, r_2, g_2, b_2)$ in I'; and compute the differences (d_x, d_y, d_z, d_r, d_g, d_b) between V_1 and V_2 by the following formulas:

$$d_x = |x_1 - x_2|, d_y = |y_1 - y_2|, d_z = |z_1 - z_2|,$$

$$d_r = |r_1 - r_2|, d_g = |g_1 - g_2|, d_b = |b_1 - b_2|.$$

6.2. If $d_\mathrm{x} < 8$, $d_\mathrm{y} < 8$, $d_\mathrm{z} < 8$, $d_\mathrm{r} < 2$, $d_\mathrm{g} < 2$, $d_\mathrm{b} < 2$, then take 7 bits M from the recovery string S and hide them into the pair of V_1 and V_2 by the following steps; otherwise, go to Step 6.1.

6.2.1. Transform d_x into a binary string T_x and shift left two bits of T_x.

6.2.2. Take two bits from M in order and embed them into the two LSBs of T_x.

6.2.3. Transform the 5-bit string T_x into an integer d_x', and *compute* the average value l_x between (x_1, x_2) by $l_x = \left\lfloor \frac{x_1 + x_2}{2} \right\rfloor$.

6.2.4. If $x_1 > x_2$, then compute $x_1' = l_x + \left\lfloor \frac{d_x' + 1}{2} \right\rfloor$ and $x_2' = l_x - \left\lfloor \frac{d_x'}{2} \right\rfloor$; otherwise, compute $x_1' = l_x - \left\lfloor \frac{d_x'}{2} \right\rfloor$ and $x_2' = l_x +$

$\left\lfloor \dfrac{d_x{}' + 1}{2} \right\rfloor$.

6.2.5. Perform the process of Steps 6.2.1 through 6.2.4 above *to* y_1 and y_2 in a similar way.

6.2.6. Transform d_r into a binary string T_r and shift left one bit *of* T_r.

6.2.7. Take the one bit from M in order to embed into the LSB of T_r.

6.2.8. Transform the 2-bit string T_r into an integer d_r', and *compute* the average value l between (r_1, r_2) by $l_r = \left\lfloor \dfrac{x_1 + x_2}{2} \right\rfloor$.

6.2.9. Perform the process above of Steps 6.2.6 through 6.2.9 to g_1, g_2, b_1 and b_2 in a similar way.

6.2.10. Take (x_1', y_1', z_1', r_1', g_1', b_1') and (x_2', y_2', z_2', r_2', g_2', b_2') as the new values of V_1 and V_2, respectively.

Step 7. If there remain unprocessed bits in S, then go to Step 6.1; otherwise, take the processed I' as the desired image I' with the recovery information S hidden.

5.3.2 Extraction of Information for Recovery

In the extraction process, at first we partition the watermarked 3D image I' into pairs with each pair consisting of two neighboring points and the pairing being done horizontally. Then, in a raster-scan order, we index each pair in the watermarked 3D image I'. Next, we use a secret key K and a random number generator f_r to generate a random number sequence H. According to the order of the random sequence H, we can take an index i from H and use it to find a pair of two neighboring points V_1' (x_1', y_1', z_1', r_1', g_1', b_1') and V_2' (x_2', y_2', z_2', r_2', g_2', b_2') with V_1' being with the index i. Then, we extract the hidden information in V_1' (x_1', y_1', z_1', r_1', g_1', b_1') and V_2' (x_2', y_2', z_2', r_2', g_2', b_2') by an inverse version of the difference expansion method proposed by Tian [6]. Furthermore, we extract the hidden bits according to the difference between V_1' and V_2'. For example, according to the difference d_r' between r_1' and r_2', we transform d_r' into a binary string T_r and extract the LSB of T_r, which is the hidden bit b'. Then, we change the difference d_r' into the original difference d_r by dividing d_r' by two and taking the floor of the result. According to d_r and the floor of the average value of r_1' and r_2', we can recover r_1' and r_2' to be r_1 and r_2, which are the original values before embedding the hidden bit, as proved in [6] . In this way, we can extract all the hidden bits in the watermarked 3D image I' for use in the later process of recovering the covered region.

The detail of the proposed process for extracting the recovery information in the 3D watermarking image I' is described as an algorithm, Algorithm 5.3.2, below.

Algorithm 5.3.2: extraction of the information for recovery.

Input: the 3D watermarking image I' with the recovery information embedded, a random number generator f, and a secret key K.

Output: a string S_r of the recovery information.

Steps:

Step 1. Use the random number generator f and the secret key K to generate a random sequence H.

Step 2. According to the order of the random numbers in H, find a pair of two neighboring points V_1' (x_1', y_1', z_1', r_1', g_1', b_1') and V_2' (x_2', y_2', z_2', r_2', g_2', b_2') in the 3D watermarking image I' as described previously; and compute the differences (d_x', d_y', d_z', d_r', d_g', d_b') between V_1' and V_2' by the following formulas:

$$d_x = |x_1 - x_2|, d_y = |y_1 - y_2|, d_z = |z_1 - z_2|,$$

$$d_r = |r_1 - r_2|, d_g = |g_1 - g_2|, d_b = |b_1 - b_2|.$$

Step 3. If $d_{\mathrm{x}}' < 32$, $d_{\mathrm{y}}' < 32$, $d_{\mathrm{z}}' < 8$, $d_{\mathrm{r}}' < 4$, $d_{\mathrm{g}}' < 4$ and $d_{\mathrm{b}}' < 4$, then go to the next step; otherwise, go to Step 2 find the next pair of points in I'.

Step 4. For the pair of (x_1', x_2'), transform d_{x}' into a binary string T_x, extract the two LSBs of T_x, and append the two bits to the extracted string M.

Step 5. Compute the average value l_x of x_1' and x_2' by $l_x = \left\lfloor \frac{x_1' + x_2'}{2} \right\rfloor$, and the new difference d_x as $d_x = \left\lfloor \frac{d_x'}{4} \right\rfloor$.

Step 6. If $x_1' > x_2'$, then compute $x_1 = l_x + \left\lfloor \frac{d_x + 1}{2} \right\rfloor$ and $x_2 = l_x - \left\lfloor \frac{d_x}{2} \right\rfloor$; otherwise, compute $x_1 = l_x - \left\lfloor \frac{d_x}{2} \right\rfloor$ and $x_2 = l_x + \left\lfloor \frac{d_x + 1}{2} \right\rfloor$.

Step 7. Perform Step 3 through Step 5 to the pair of (y_1', y_2').

Step 8. For the pair of (r_1', r_2'), transform d_{r}' into a binary string T_r, extract the LSB of T_r, and append the bit to the extracted string M.

Step 9. Compute the average value l_r of r_1' and r_2' by $l_r = \left\lfloor \frac{r_1' + r_2'}{2} \right\rfloor$ and the new difference d_r as $d_r = \left\lfloor \frac{d_r'}{2} \right\rfloor$.

Step 10. If $r_1' > r_2'$, then compute $r_1 = l_r + \left\lfloor \frac{d_r + 1}{2} \right\rfloor$ and $r_2 = l_r - \left\lfloor \frac{d_r}{2} \right\rfloor$; otherwise, compute $r_1 = l_r - \left\lfloor \frac{d_r}{2} \right\rfloor$ and $r_2 = l_r + \left\lfloor \frac{d_r + 1}{2} \right\rfloor$.

Step 11. Perform the Step 7 through Step 9 to the pairs of (g_1', g_2') and (b_1', b_2'), respectively.

Step 12. If there remain unprocessed pair in I', then go to Step 2; otherwise, take the string M as the desired string S_r.

5.3.3 Eliminating 3D Watermarks and Recovery of KINECT Images

In the proposed process for eliminating the 3D watermark and recovering the original KINECT image, at first we use the secret key K to resume the order of the extracted string S_r. Then, according to the 3D watermark W, we compute the covered region in the watermarked 3D image I'. Next, we find the position of the covered region in the raster-scan order, transform S_r into the original coordinates (x, y, z, r, g, b) in order, and recover the value of the covered region with the original coordinates. In this way, we can recover the covered region with the original 3D image data and remove the 3D watermark from the watermarked 3D image I'.

The detail of the proposed process for eliminating the 3D watermark in the watermarked 3D image I' and recovering the covert region of the 3D watermark is described as an algorithm, Algorithm 5.3.3, below.

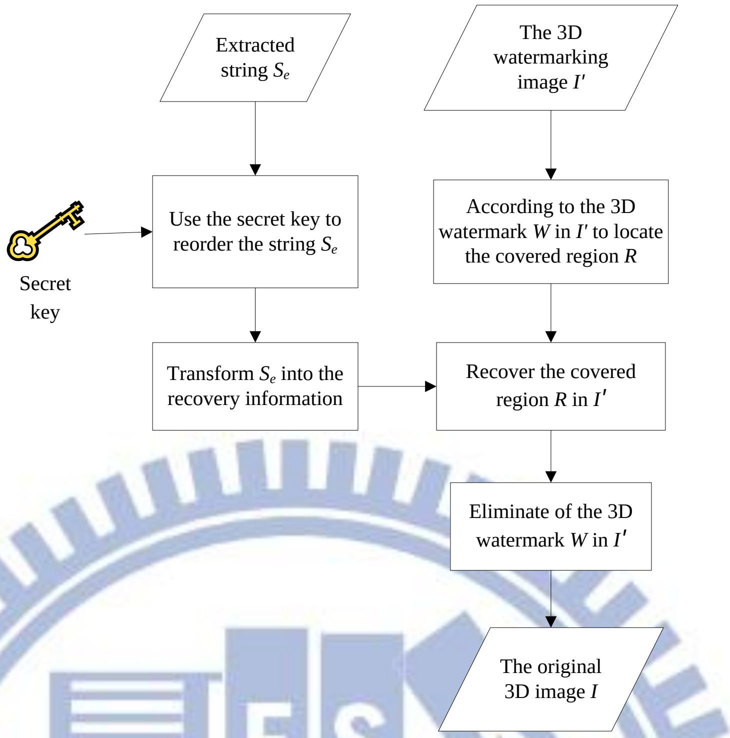

Extracted string S_e
The 3D watermarking image I'
Use the secret key to reorder the string S_e
According to the 3D watermark W in I' to locate the covered region R
Secret key
Transform S_e into the recovery information
Recover the covered region R in I'
Eliminate of the 3D watermark W in I'
The original 3D image I

Figure 5.3 Illustration of eliminating the 3D watermark and recovering the covered region in the 3D image.

Algorithm 5.3.3: eliminating the 3D watermark and recovering the covert region.

Input: the watermarked 3D image I' with the recovery information embedded, a secret key K and a string S_r includes the recovery information.

Output: the original 3D image I_r.

Steps:

Step 1. Use the secret key K to reorder the string S_r.

Step 2. Extract 37 bits from string S in order, and transform them into the format (r, g, b, d)

Step 3. In a raster-scan order, find the covered region R of the 3D watermarking image I' by the following formulas:

$$u_c = x_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}}, \quad v_c = y_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}},$$

where (x_w, y_w, z_w) are the coordinates of the 3D watermark W in I' and (u_c, v_c) are the coordinates of the covered region in I'.

Step 4. According to (u_c, v_c), transform the extracted (r, g, b, d) into (x, y, z, r, g, b) in order by the following formulas:

$$x = u_c \cdot \frac{d}{\sqrt{u_c^{\,2} + v_c^{\,2} + f^{\,2}}},\ y = v_c \cdot \frac{d}{\sqrt{u_c^{\,2} + v_c^{\,2} + f^{\,2}}},\ z = f \cdot \frac{d}{\sqrt{u_c^{\,2} + v_c^{\,2} + f^{\,2}}},$$

where f is the focal length of the IR camera in the KINECT device.

Step 5. Use the recovery information (x, y, z, r, g, b) to recover the covered region in the watermarked 3D image I' by replacing the covered region with the corresponding recovery information (x, y, z, r, g, b).

Step 6. If there remain bits in S, then go to Step 2; otherwise, go to Step 7.

Step 7. Eliminate the 3D watermark by removing the data of the 3D watermark in I' and take the processed I' as the desired image I_r.

5.3.4 Experimental results

At the beginning of the experiment, the result of combining the original KINECT images, which include a depth image and a color image, to be a 3D image I by the proposed process is shown in Figure 5.4. And the input 3D visible watermark is shown in Figure 5.5. Then, we embed the 3D visible watermark into the 3D image I and hide the information of covered region into the watermarked 3D image I' by Algorithm 5.3.1. The result of the watermarked 3D image I' is shown in Figure 5.6 and the result of hiding the recovery information into I' by the difference expansion method is shown in Figure 5.7. Then, we extract the hidden data string S by Algorithm 5.3.2 and use the secret key K to resume the order of extracted string S. By using the string S, we can recover the covered region and eliminate the 3D watermark from the watermarked 3D image I'. The result of eliminating the 3D watermark and recover of the covered region is shown in Figure 5.8.

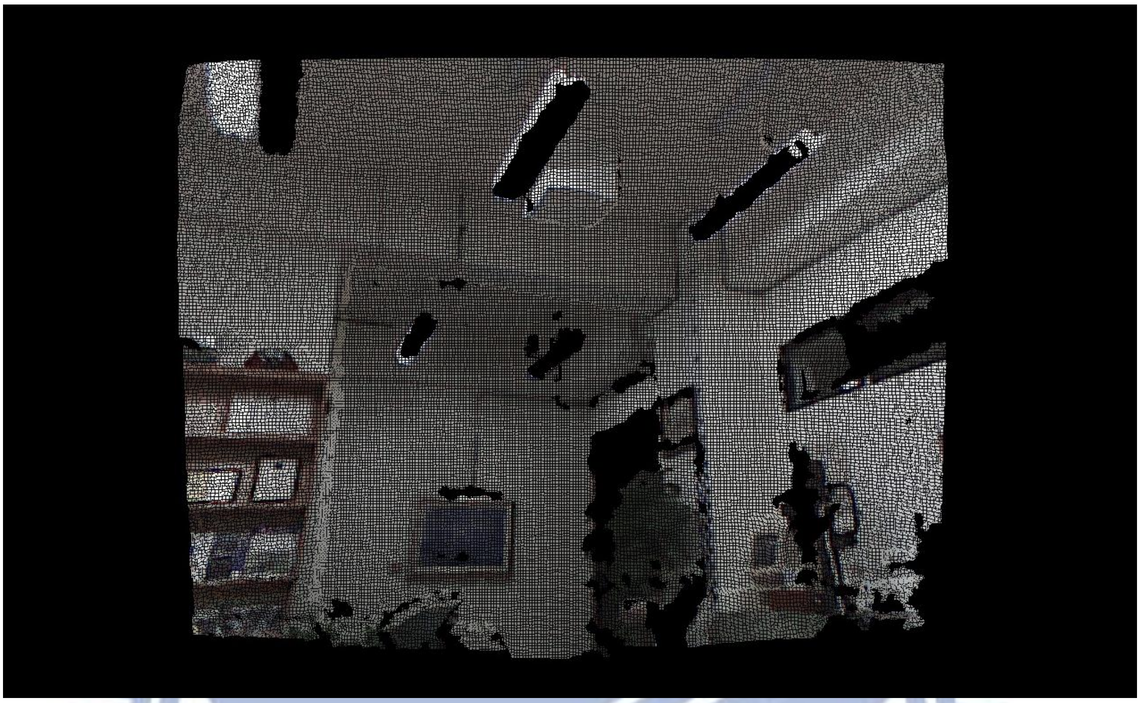


Figure 5.4 The result of combining the original depth and color image into a 3D image.

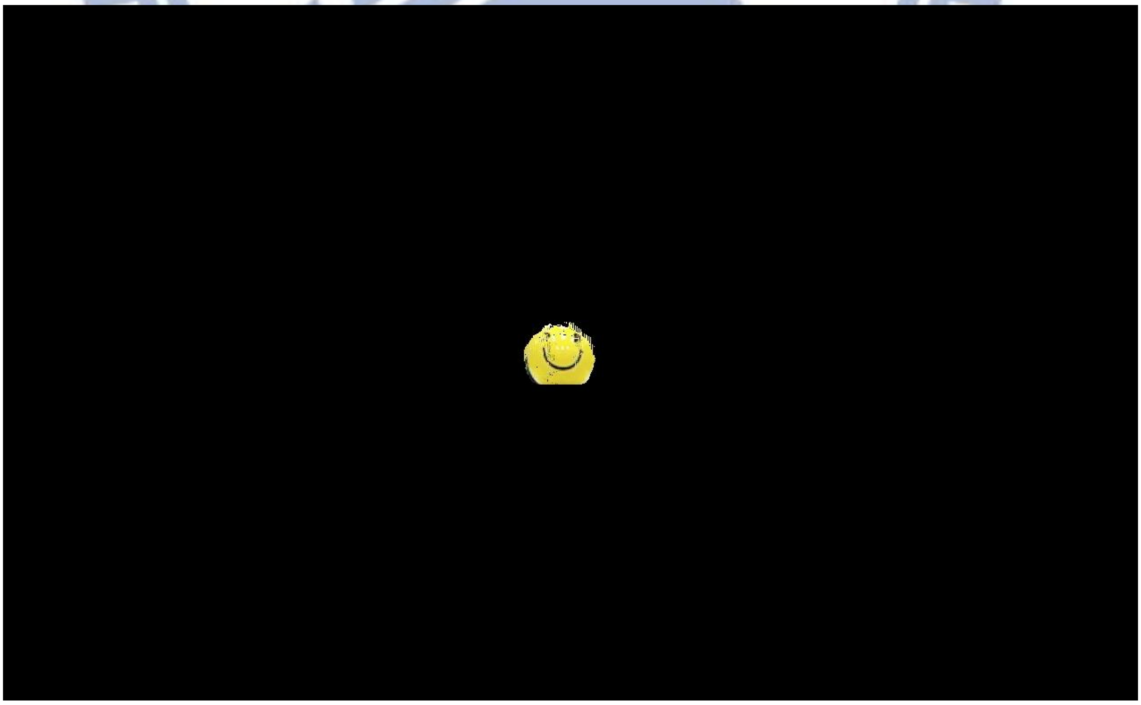


Figure 5.5 The 3D visible watermark.

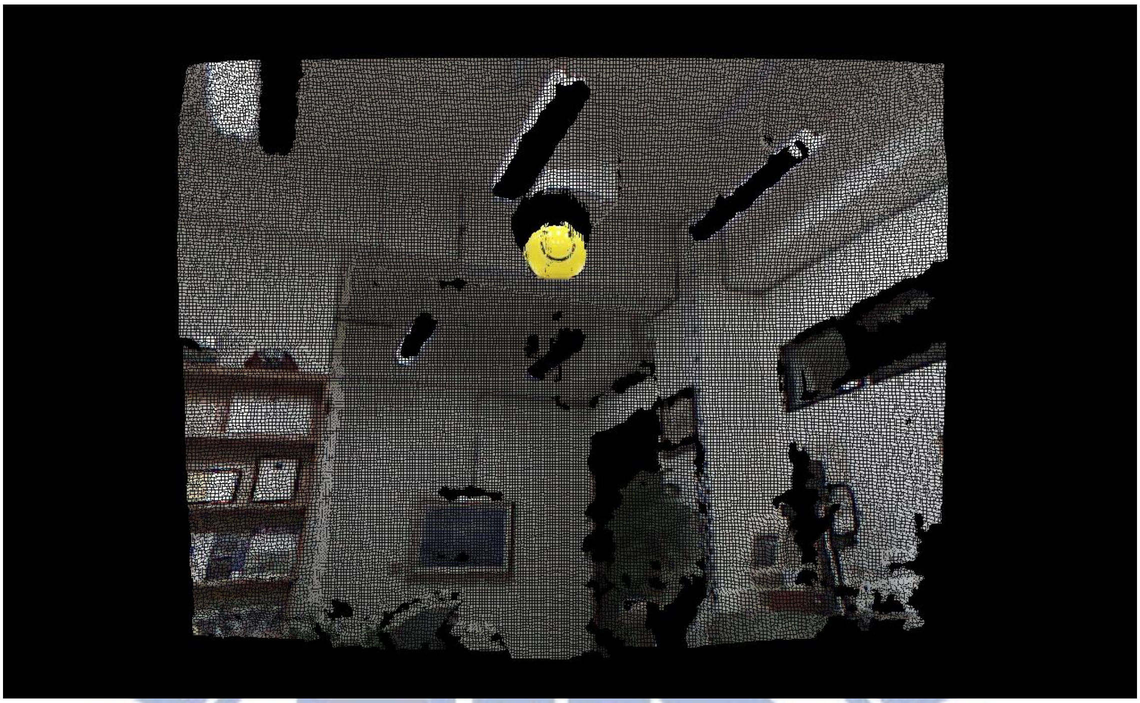


Figure 5.6 The result of the watermarked 3D image.

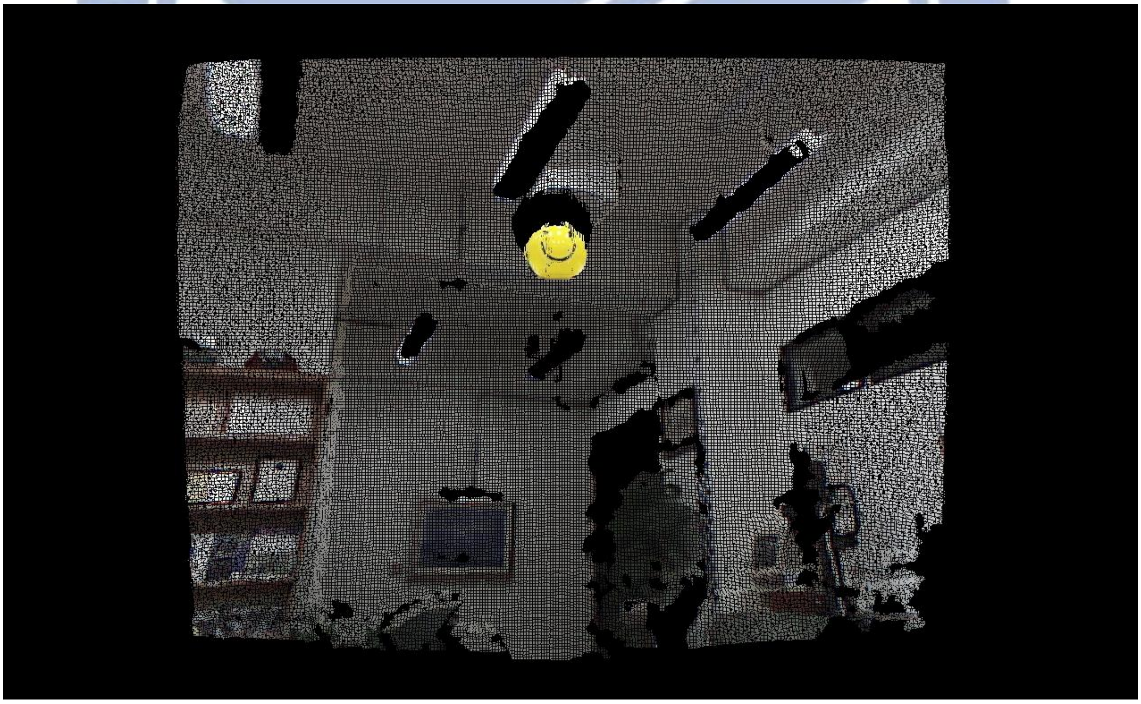


Figure 5.7 The watermarked 3D image in which the recovery data string is embedded by the difference expansion method.

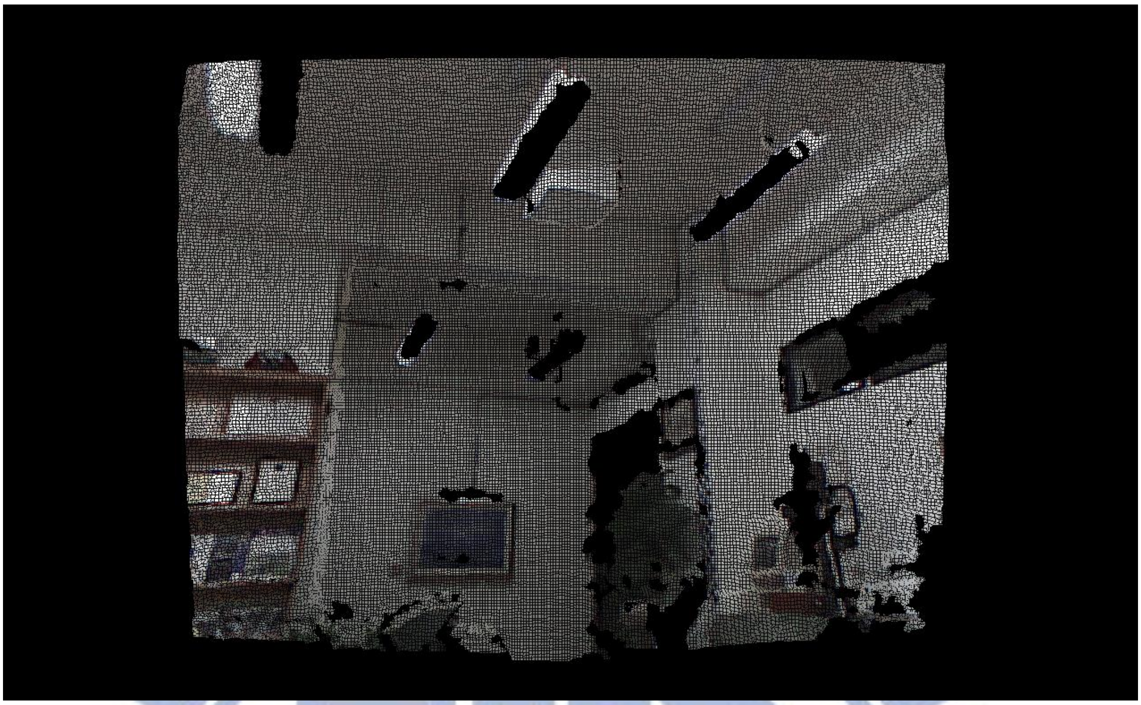


Figure 5.8 The result of eliminating the 3D watermark and recovering the covered region.

5.4 Embedding and Eliminating 3D Visible Watermarks in KINECT Images by Reversible Contrast Mapping

In this section, the details of the proposed method for embedding and eliminating a 3D visible watermark in a 3D image constructed from KINECT images (color and depth images) by the reversible contrast mapping method are described. The detailed process of embedding a 3D visible watermark into a 3D image and hiding the recovery information in the watermarked 3D image is described in Section 5.4.1. In Section 5.4.2, the process of extracting the recovery information from the watermarked 3D image is described. And in Section 5.4.3, the process of eliminating the 3D visible watermark and recovering the covered region is described. Some experimental results of applying the proposed algorithms are given in Section 5.4.4.

5.4.1 Embedding of 3D Visible Watermarks

In the process of embedding of a 3D visible watermark into a 3D image (a color image or a depth image), at first we transform every pixel value in the depth image into the corresponding 3D real-world coordinates (x, y, z). Then, according to these coordinates, we try to find the corresponding color information (r, g, b) for each pixel. In this way, the original depth and color images are combined together into a 3D image I, which includes all the information of the KINECT images, with its coordinates formatted as (x, y, z, r, g, b).

In addition, the information of the 3D visible watermark is also formatted as (x, y, z, r, g, b), such that we can embed the 3D visible watermark W into the 3D image I. Then, the 3D visible watermark W will cover some region of the original 3D image I. And we transform the data, which are covered by the 3D visible watermark W, into a recovery-data string S and use a secret key K to randomize the order of the bits in the string S. Finally, we hide the string S into the watermarked 3D image I' by the reversible contrast mapping method proposed by Coltuc and Chassery [8].

During the hiding process, at first we partition the watermarked 3D image I' into point pairs. Each point pair consists of two neighboring points and the pairing is done horizontally. Then, in a raster-scan order, we index each pair in the watermarked 3D image I'. Next, we use a secret key K and a random number generator f_r to generate a random number sequence H. According to the order of the random sequence H, we use an index i of H to find a pair of two neighboring points $V_1(x_1, y_1, z_1, r_1, g_1, b_1)$ and $V_2(x_2, y_2, z_2, r_2, g_2, b_2)$ with V_1 being with the index i.

In addition, according to the different range of values between the 3D coordinates (x, y, z) and the color information (r, g, b), two domains D_1 and D_2 are defined. And if a pair is in the *corresponding* domain, we hide the bit, which is taken from string S, into this pair; otherwise, we do not hide the bit in this pair. Specifically, for the 3D coordinate pair (x_1, x_2), the domain D_1 is defined by $0 \le 2x_1 - x_2 \le 8000$ and $0 \le 2x_2 - x_1 \le 8000$. And for the color information pair (r_1, r_2), the domain D_2 is defined by $0 \le 2r_1 - r_2 \le 255$ and $0 \le 2r_2 - r_1 \le 255$. The restrictions on the domains D_1 and D_2 prevent occurrences of overflows or/and underflows when data transformations during data hiding are conducted. When we hide a bit into a pair, three conditions will be encountered, as explained next.

For a 3D coordinate pair (x_1, x_2), (a) if (x_1, x_2)$\in D_1$ and if either x_1 and x_2 is not an odd value , then we transform (x_1, x_2) into (x_1', x_2') by $x_1' = 2x_1 - x_2$ and $x_2' = 2x_2 - x_1$, set the LSB of x_1' to be “1,” and hide the bit in the LSB of x_2'; (b) if (x_1, x_2) $\in D_1$ and if both x_1 and x_2 are odd values, then we set the LSB of x_1 to “0” and hide the bit in the LSB of x_2; (c) if (x_1, x_2) $\notin D_1$, set the LSB of x_1 to “0” and append the original LSB of x_1 to the end of the data string for hiding in order to losslessly recover the original LSB of x_1 later during the extraction process. In this way, we can hide the string S into the watermarked 3D image I'. As for a color information pair (r_1, r_2), its use of data embedding is similar to the use of the 3D coordinate pair (x_1, x_2) described above except that the domain D_2 is used rather than D_1.

For instants, (a) if a pair (r_1, r_2) = (3, 6) and a bit “1” taken from the string to be hidden, then because $0 \le 2r_1 - r_2 = 0 \le 255$ and $0 \le 2r_2 - r_1 = 9 \le 255$, we get to know that ($r_1$, r_2) $\in D_2$. Also, r_2 is not an odd value, so we transform (r_1, r_2) into (r_1', r_2') by $r_1' = 2r_1 - r_2 = 0$, and $r_2' = 2r_2 - r_1 = 9$; and set the LSB of r_1' to be “1,” and hide the bit “1” in the LSB of r_2', such that we take the pair (r_1', r_2') = (1, 9) as the final result of data embedding. (b) If a pair (r_1, r_2) = (7, 5) and a bit “1” is taken from the string to be hidden, then because $0 \le 2r_1 - r_2 = 9 \le 255$ and $0 \le 2r_2 - r_1 = 3 \le 255$, we get to know that ($r_1$, r_2)$\in D_2$. Also, both r_1 and r_2 are odd values, so we set the LSB of r_1 to “0” and hide the bit “1” in the LSB of r_2, such that we take the pair (r_1', r_2') = (6, 5) as the final result of data embedding. (c) If a pair (r_1, r_2) = (2, 5), because $2r_1 - r_2 = -1 < 0$, we get to know that (r_1, r_2)$\notin D_2$. So, we set the LSB of r_1 to be "0" and append the original LSB of r_1 to the end of the data string for hiding, such that we take the pair (2, 5) as the final result of data embedding.

The detail of the above proposed process for embedding a 3D visible watermark is presented as an algorithm, Algorithm 5.4.1, below.

Algorithm 5.4.1: embedding of the 3D watermark.

Input: a KINECT image, which includes a depth image and a color image, a secret key *K*, a random number generator *f*, and a 3D visible watermark *W*.

Output: a 3D watermarked image *I'* with the recovery information embedded.

Steps:

Step 1. Transform the depth image and the color image into a 3D image *I* which is *formatted* as (*x*, *y*, *z*, *r*, *g*, *b*) by the transformation process described in Section 5.2.

Step 2. According to the positions described by (x_w, y_w, z_w, r_w, g_w, b_w) of all the pixels of *W*, deicide a region *R* in *I* whose pixels' coordinates are computed by the following formulas which are derived in a way similar to that discussed in Section 5.2:

$$u_c = x_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}}, \quad v_c = y_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}},$$

where (x_w, y_w, z_w) are the pixels' coordinates of the 3D watermark *W* and (u_c, v_c) are the pixels' coordinates of the region *R* in the 3D image *I* covered by the 3D watermark *W*.

Step 3. Transform the data (*x*, *y*, *z*, *r*, *g*, *b*) at the position (u_r, v_r) in the covered region *R* into (*d*, *r*, *g*, *b*) by replacing (*x*, *y*, *z*) with the depth value *d* of the depth pixel at (u_r, v_r) in the depth image for reducing the data that need be hidden; and then transform every (d, r, g, b) into a recovery binary string S by the following steps: (1) transform each d into a 13-bit string S_1; (2) transform each of r, g, and b into three 8-bit strings S_2, S_3, and S_4; (3) combine S_1 through S_4 into a string T; and (4) append T to string S.

Step 4. Embed the 3D watermark W into the 3D image I by replacing the data of region R in I with the data in W.

Step 5. Use the secret key K to randomize the order of string S and use the random number generator f and secret key K to generate a random sequence H.

Step 6. Hide the recovery string S into the 3D watermarking image I' by the reversible contrast mapping method by Coltuc and Chassery [8] as follows.

6.1 In the order of H, take a number i from H and find a pair of two neighboring points $V_1(x_1, y_1, z_1, r_1, g_1, b_1)$ and V_2,($x_2, y_2, z_2, r_2, g_2, b_2$) in I' with V_1 being indexed by i.

6.2 If $0 \le x_1' \le 8000$, $0 \le x_2' \le 8000$ where

$$x_1' = 2x_1 - x_2\,,\quad x_2' = 2x_2 - x_1 \tag{1}$$

and if either x_1 and x_2 is not an odd value, then transform (x_1, x_2) into (x_1', x_2') by (1) above, set the LSB of x_1' to be “1,” and take one bit from string S and embed it into the LSB of x_2'.

6.3 If $0 \le x_1' \le 8000$, $0 \le x_2' \le 8000$ and if both x_1 and x_2 are odd values, set the LSB of x_1 to be “0,” and take one bit from string S and embed it into the LSB of x_2.

6.4 If $x_1' > 8000$ or $x_1' < 0$ and $x_2' > 8000$ or $x_2' < 0$, set the LSB of x_1 to be “0,” and append the original LSB of x_1 to the end of string S.

6.5 Perform the process of Steps 6.2 through 6.4 above to y_1, y_2 and z_1, z_2 in a similar way.

6.6 If $0 \le r_1' \le 255$, $0 \le r_2' \le 255$ where

$$r_1' = 2r_1 - r_2,\ r_2' = 2r_2 - r_1, \tag{2}$$

and if either r_1 and r_2 is not an odd value, transform (r_1, r_2) by (2), set the LSB of r_1' to be "1," and take one bit from string S and embed it into the LSB of r_2'.

6.7 If $0 \le r_1' \le 255$, $0 \le r_2' \le 255$ and if both r_1 and r_2 are odd values, set the LSB of r_1 to be "0," and take one bit from string S and embed it into the LSB of r_2.

6.8 If $r_1' > 255$ or $r_1' < 0$ and $r_2' > 255$ or $r_2' < 0$, set the LSB of r_1 to be "0," and append the original LSB of r_1 to the end of string S..

6.9 Perform the process of Steps 6.6 through 6.8 above to g_1, g_2 and b_1, b_2 in a similar way.

Step 7. If there remain unprocessed bits in S, then go to Step 6.1; otherwise, take the processed I' as the desired image I' with the recovery information S hidden.

5.4.2 Extraction of Information for Recovery

In the extraction process, at first we partition the watermarked 3D image I' into pairs with a pair consisting of two neighboring points and the pairing is done horizontally. Then, in a raster-scan order, we index each pair in the watermarked 3D image I'. Next, we use a secret key K and a random number generator f_r to generate a random number sequence H. In the order of the random sequence H, we take a number i from H and find accordingly a pair of two neighboring points $V_1'(x_1', y_1', z_1', r_1', g_1', b_1')$ and $V_2'(x_2', y_2', z_2', r_2', g_2', b_2')$ with V_1' being indexed by i. Then, we extract the hidden information in $V_1'(x_1', y_1', z_1', r_1', g_1', b_1')$ and $V_2'(x_2', y_2', z_2', r_2', g_2', b_2')$ by the inverse of reversible contrast mapping method proposed by Coltuc and Chassery [8]. In this way, we can extract all the recovery information in *I'*.

For example, in the pair (x_1', x_2'), (a) if the LSB of x_1' is “1,” then we can extract the hidden bit from the LSB of x_2' and, in addition, recover the original pair (x_1, x_2) by the inverse transformation $x_1 = \left\lceil \frac{2}{3}x_1' + \frac{1}{3}x_2' \right\rceil$, $x_2 = \left\lceil \frac{1}{3}x_1' + \frac{2}{3}x_2' \right\rceil$; (b) if the LSB of x_1' is “0” and the pair (x_1', x_2') with the LSBs set to “1” belongs to D_1, then we can extract the hidden bit from the LSB of x_2'; (c) if the LSB of x_1' is “0” and the pair (x_1', x_2') with the LSBs set to “1” does not belong to D_1, then we recover the LSB of x_1' by replacing the LSB of x_1' with the corresponding original value extracted from the extraction sequence.

The detail of the proposed process for extraction the recovery information in the 3D watermarking image *I'* is described as an algorithm, Algorithm 5.3.2, below.

Algorithm 5.4.2: extraction of the hidden data.

Input: the 3D watermarked image *I'* with the recovery information embedded, a random number generator *f*, and a secret key *K*.

Output: a string S_r of recovery information.

Steps:

Step 1. Use the random number generator *f* and secret key *K* to generate a random sequence *H*.

Step 2. In the order of *H*, take out a number *i* in *H* and find accordingly a pair of two neighboring points $V_1'(x_1', y_1', z_1', r_1', g_1', b_1')$ and $V_2'(x_2', y_2', z_2', r_2', g_2', b_2')$ in the 3D watermarked image *I'* with V_1' being indexed by *i*.

Step 3. If the LSB of x_1' is “1,” then extract the LSB of x_2', append it to the extracted string *S*, set the LSBs of x_1' and x_2' to be “0,” and recover the original pair (x_1, x_2) by

$$x_1 = \left\lceil \frac{2}{3}x_1' + \frac{1}{3}x_2' \right\rceil, \;\; x_2 = \left\lceil \frac{1}{3}x_1' + \frac{2}{3}x_2' \right\rceil.$$

Step 4. If the LSB of x_1' is “0” and the pair (x_1', x_2') with the LSBs of x_1' and x_2' set to be “1” satisfies $0 \le x_1 \le 8000$, $0 \le x_2 \le 8000$, where $x_1 = 2x_1' - x_2'$, $x_2 = 2x_2' - x_1'$, then extract the LSB of x_2' and append it to the extracted string S, and restore the original pair (x_1', x_2') with the LSBs of x_1' and x_2' set to “1.”

Step 5. If the LSB of x_1' is “0” and the pair (x_1', x_2') with the LSBs of x_1' and x_2' set to “1” does not satisfy $0 \le x_1 \le 8000$, $0 \le x_2 \le 8000$, where

$$x_1 = 2x_1' - x_2', x_2 = 2x_2' - x_1',$$

then recover the original (x_1, x_2) by replacing the LSB of x_1' with the corresponding original value extracted from the extracted string S.

Step 6. Perform the process of Steps 3 through 5 above to y_1, y_2 and z_1, z_2 in a similar way.

Step 7. If the LSB of r_1' is “1,” then extract the LSB of r_2', append it to the extracted string S, set the LSBs of r_1' to be “0,” and recover the original pair (r_1, r_2) by

$$r_1 = \left\lceil \frac{2}{3}r_1' + \frac{1}{3}r_2' \right\rceil, \;\; r_2 = \left\lceil \frac{1}{3}r_1' + \frac{2}{3}r_2' \right\rceil.$$

Step 8. If the LSB of r_1' is “0” and the pair (r_1', r_2') with the LSBs of r_1' and r_2' set to be “1” satisfies $0 \le r_1 \le 255$, $0 \le r_2 \le 255$, where

$$r_1 = 2r_1' - r_2', r_2 = 2r_2' - r_1',$$

then extract the LSB of r_2' and add it to the extracted string S, and restore the original pair (r_1', r_2') with the LSBs of r_1' and r_2' set to “1”.

Step 9. If the LSB of r_1' is “0” and the pair (r_1', r_2') with the LSBs of r_1' and r_2' set to be “1” does not satisfy $0 \le r_1 \le 255$, $0 \le r_2 \le 255$, where

$$r_1 = 2r_1' - r_2', r_2 = 2r_2' - r_1',$$

then recover the original (r_1, r_2) by replacing the LSB of r_1' with the corresponding original value extracted from the extracted string S.

Step 10. Perform the process of Steps 7 through 9 above to g_1, g_2 and b_1, b_2 in a similar way.

Step 11. If there remain unprocessed pair in I', then go to Step 2; otherwise, take the string S as the desired string S_r.

5.4.3 Eliminating 3D Watermarks and Recovery of KINECT Images

In the proposed process for eliminating the 3D watermark and recovering the original KINECT image, at first we use the secret key K to resume the order of the extracted string S_r. Then, according to the 3D watermark W, we compute the covered region in the watermarked 3D image I'. Next, we find the position of the covered region in the raster-scan order, transform S_r into the original coordinates (x, y, z, r, g, b) in order, and recover the pixels' values of the covered region with the original coordinates. In this way, we can recover the covered region with the original 3D image data and remove the 3D watermark from the watermarked 3D image I'.

The detail of the proposed process for eliminating the 3D watermark in the 3D watermarking image I' and recovering the covert region of the 3D watermark is described as an algorithm, Algorithm 5.4.3, below.

Algorithm 5.4.3: eliminating the 3D watermark and recovering the covert region.

Input: the watermarked 3D image I' with the recovery information embedded, a secret key K, and a string S_r including the recovery information.

Output: the original 3D image I_r.

Steps:

Step 1. Use the secret key *K* to reorder the string S_r.

Step 2. Extract 37 bits from string *S* in order, and transform them bits into the format of (*r*, *g*, *b*, *d*).

Step 3. In a raster-scan order, find the covered region *R* of the 3D watermarking image *I'* by the following formulas:

$$u_c = x_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}},\ v_c = y_w \cdot \frac{f}{\sqrt{{x_w}^2 + {y_w}^2 + {z_w}^2}},$$

where (x_w, y_w, z_w) are the coordinates of the 3D watermark *W* in *I'* and (u_c, v_c) are the coordinates of the covered region in *I'*.

Step 4. According to the coordinates (u_c, v_c), transform the extracted (*r*, *g*, *b*, *d*) into (*x*, *y*, *z*, *r*, *g*, *b*) in order by the following formulas:

$$x = u_c \cdot \frac{d}{\sqrt{{u_c}^2 + {v_c}^2 + f^2}}, y = v_c \cdot \frac{d}{\sqrt{{u_c}^2 + {v_c}^2 + f^2}}, z = f \cdot \frac{d}{\sqrt{{u_c}^2 + {v_c}^2 + f^2}},$$

where *f* is the focal length of the IR camera in the KINECT device.

Step 5. Use the recovery information (*x*, *y*, *z*, *r*, *g*, *b*) to recover the covered region in the watermarked 3D image *I'* by replacing the covered region with the corresponding recovery information (*x*, *y*, *z*, *r*, *g*, *b*).

Step 6. If there remain bits in *S*, then go to Step 2; otherwise, go to Step 7.

Step 7. Eliminate the 3D watermark by removing the data of the 3D watermark in *I'* and take the processed *I'* as the desired image I_r.

5.4.4 Experimental results

At the beginning of the experiment, the result of combining the original KINECT images into a 3D image *I* by the proposed process as described in Section 5.3.4 is shown in Figure 5.9. And 3D visible watermark is the same as that we use in Section 5.3.4, as shown in Figure 5.10. Then, we also embed the 3D visible watermark into the same position of 3D image *I* and hide the information of the covered region into the watermarked 3D image *I'* by Algorithm 5.4.1. The result of the watermarked 3D image *I'* is shown in Figure 5.11, and the result of hiding the recovery information into *I'* by the reversible contrast mapping method is shown in Figure 5.12. Then, we extract the hidden data string *S* by Algorithm 5.4.2 and use the secret key *K* to resume the order of extracted string *S*. By using the string *S*, we recovered the covered region and eliminate the 3D watermark from the watermarked 3D image *I'*. The result of eliminating the 3D watermark and recovering the covered region is shown in Figure 5.13.

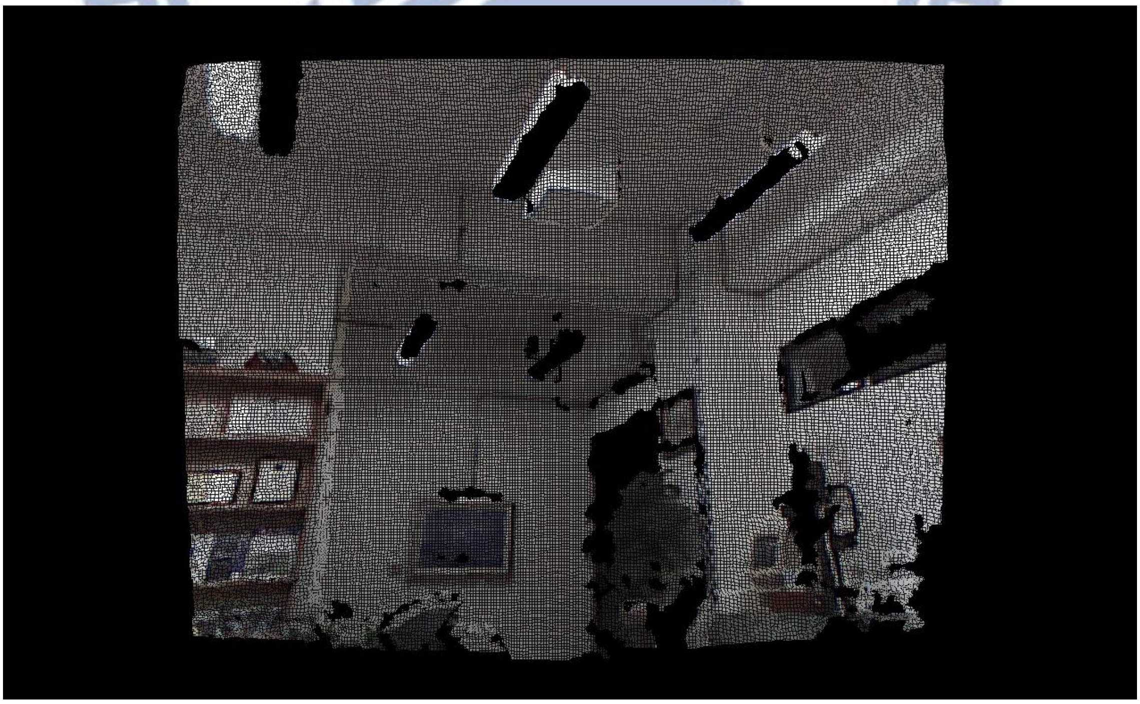


Figure 5.9 The result of combining the original depth and color image into a 3D image.

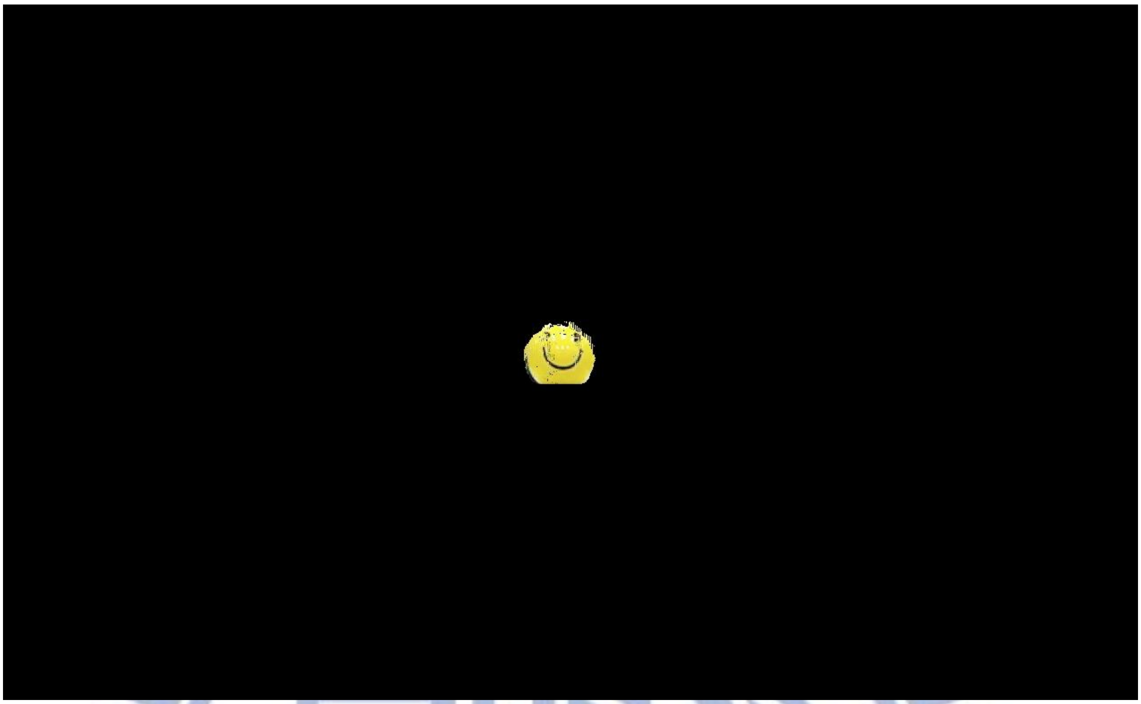


Figure 5.10 The 3D visible watermark.

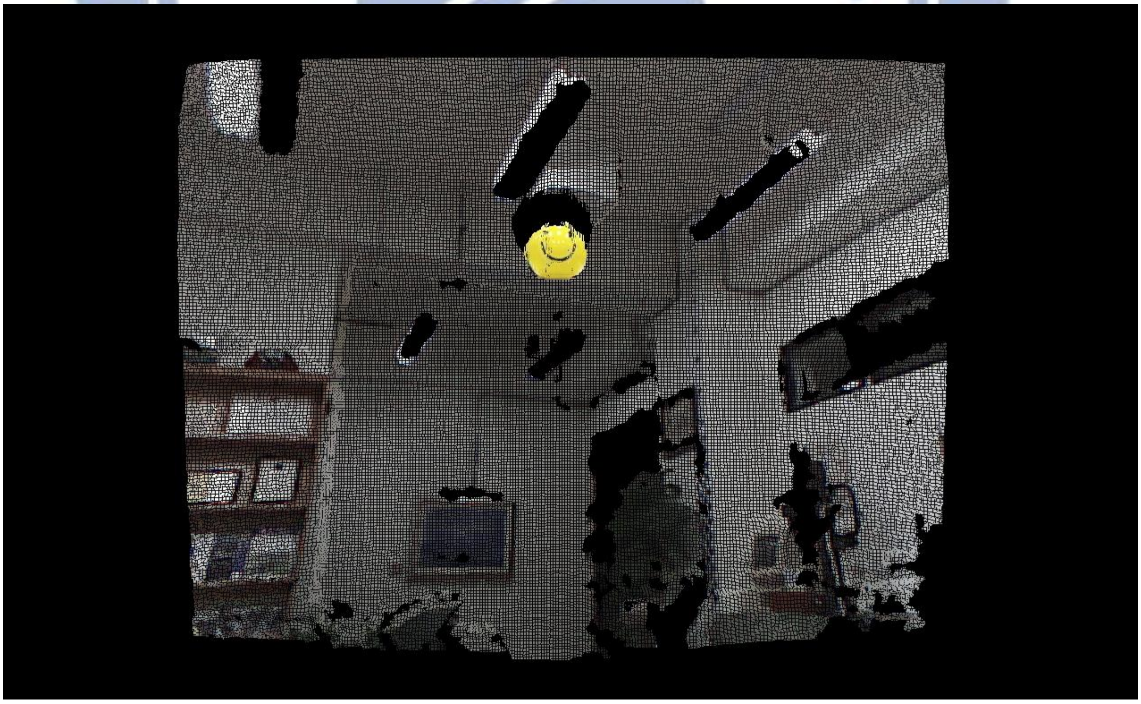


Figure 5.11 The result of the watermarked 3D image.

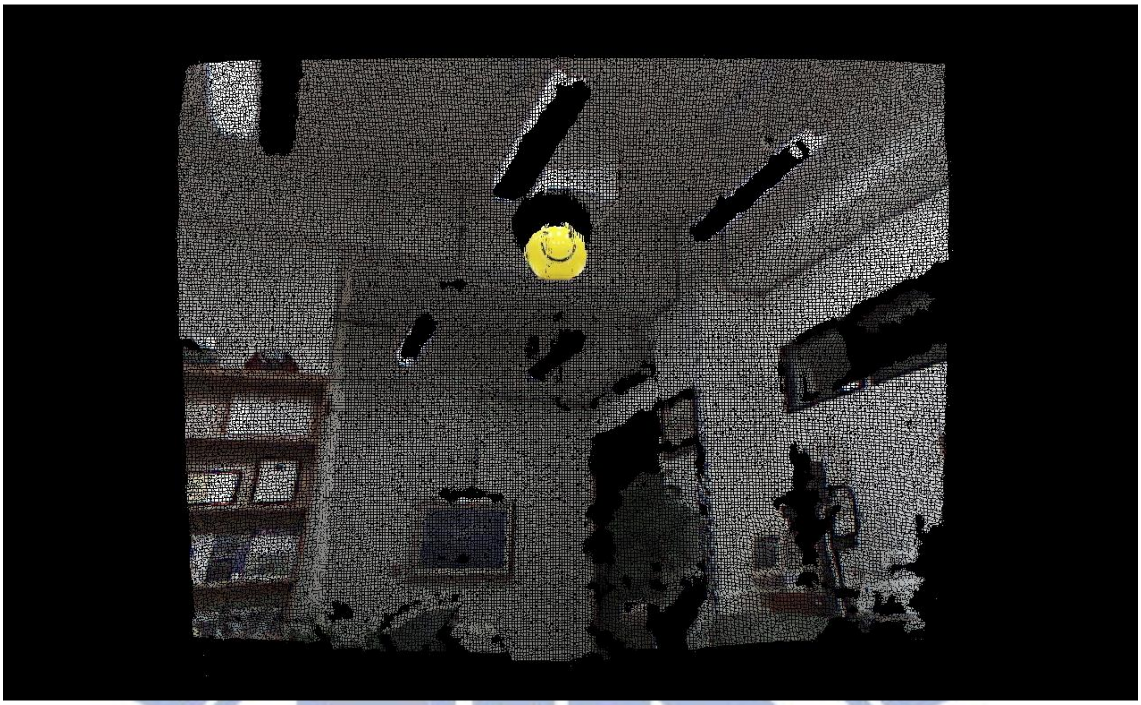


Figure 5.12 The watermarked 3D image in which the recovery data string is embedded by the reversible contrast mapping method.

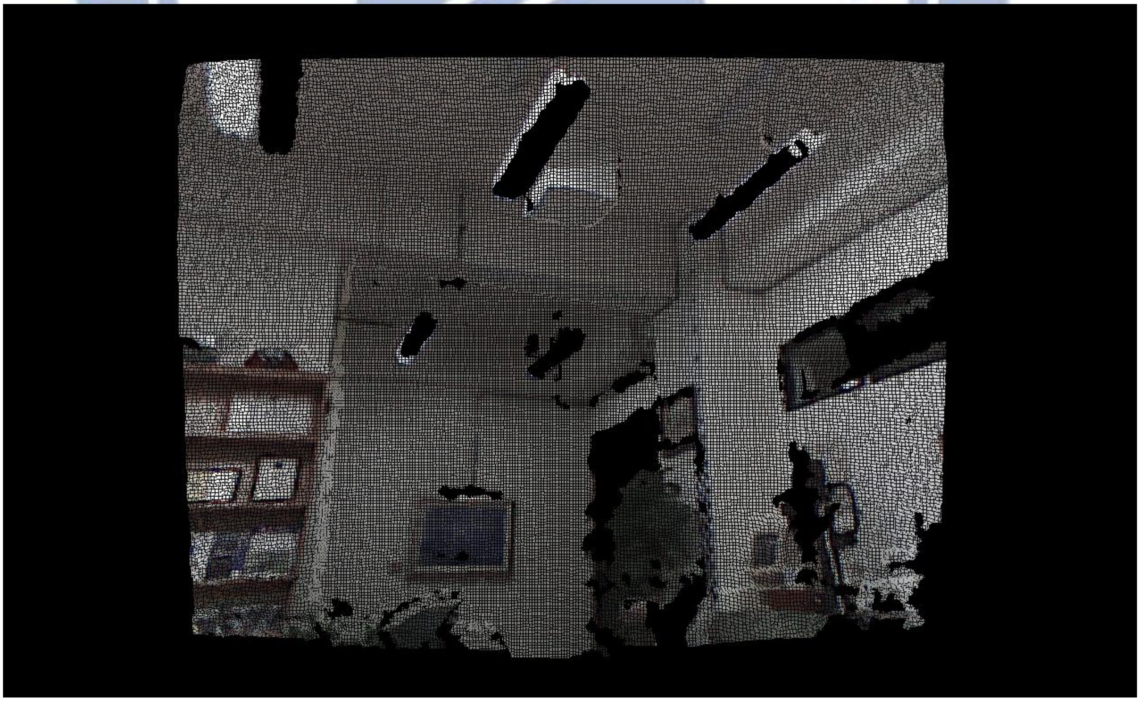


Figure 5.13 The result of eliminating the 3D watermark and recovering the covered region.

5.5 Experimental Results

In this section, some more experimental results of applying the above proposed method for copyright protection of KINECT images by 3D visible watermarking are shown in Figures 5.14 through 5.19. We will present the experimental results of the generating 3D images in the different viewpoints, and also those of embedding the 3D watermark into the different positions of the KINECT images.

At the beginning of the experiments, the result of combining the original KINECT images, which includes a depth and a color image, to form a 3D image *I* by the proposed process is shown in Figure 5.14. Then, we embedded the 3D visible watermark into the left part of 3D image *I* and hide the information of the covered region into the watermarked 3D image *I'*. The result of the watermarked 3D image *I'* is shown in Figure 5.15. In addition, the result of hiding the recovery information into *I'* by the reversible contrast mapping method is shown in Figure 5.16. Then, we extracted the hidden data string *S* by Algorithm 5.4.2 and used the secret key *K* to resume the bit order of the extracted string *S*. By using the string *S*, we recovered the covered region and eliminated the 3D watermark from the watermarked 3D image *I'*. The results of the watermarked 3D image *I'* resulting from the extraction process with the different viewpoints are shown in Figure 5.17. And the result of eliminating the 3D watermark and recovery of the covered region is shown in Figure 5.18. In addition, the result of eliminating the 3D watermark and recovery of the covered region by using the wrong secret key to extract the hidden data is shown in Figure 5.19.

We also embedded the 3D watermark into the right part of the 3D image *I*. Then, we hide the information of the covered region into the watermarked 3D image *I'* by the difference expansion method. The result of the watermarked 3D image *I'* with the recovery information embedded is shown in Figure 5.20. Then, we extracted the hidden information from the watermarked 3D image *I'* and eliminated the 3D watermark from the watermarked 3D image by the eliminating process. And the result of the 3D image resulting from the eliminating process is shown in Figure 5.21. Besides, the result of eliminating the 3D watermark and recovery of the covered region by using the wrong secret key to extract the hidden data is shown in Figure 5.22.

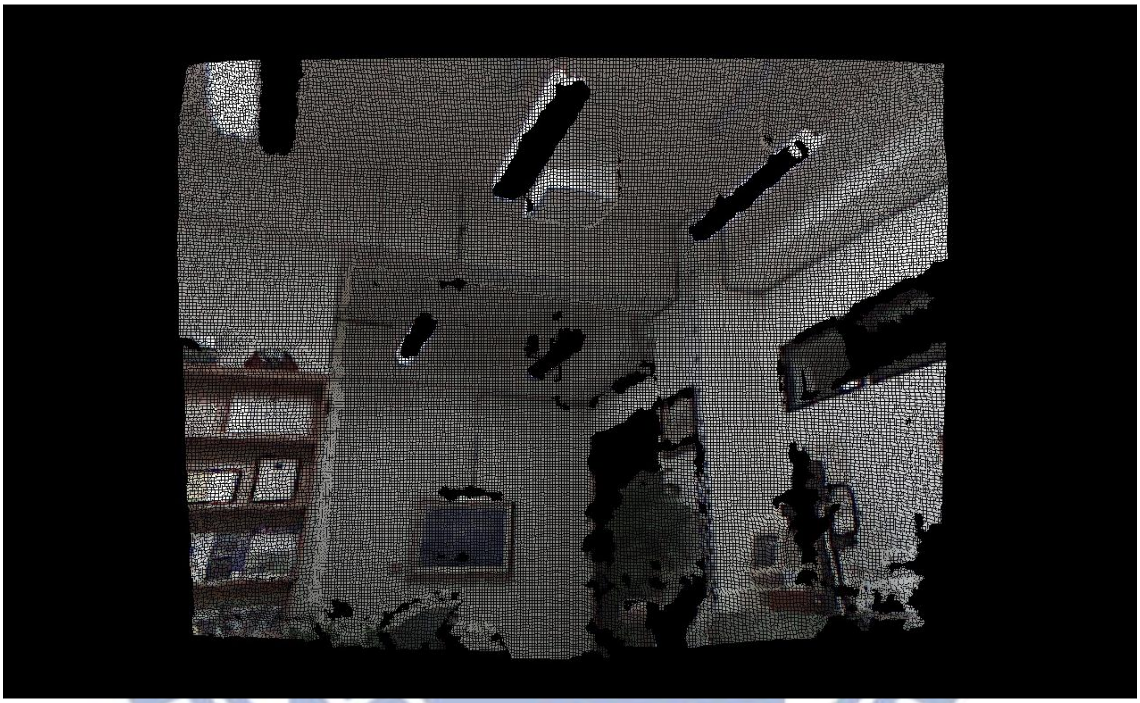


Figure 5.14 The original 3D image.

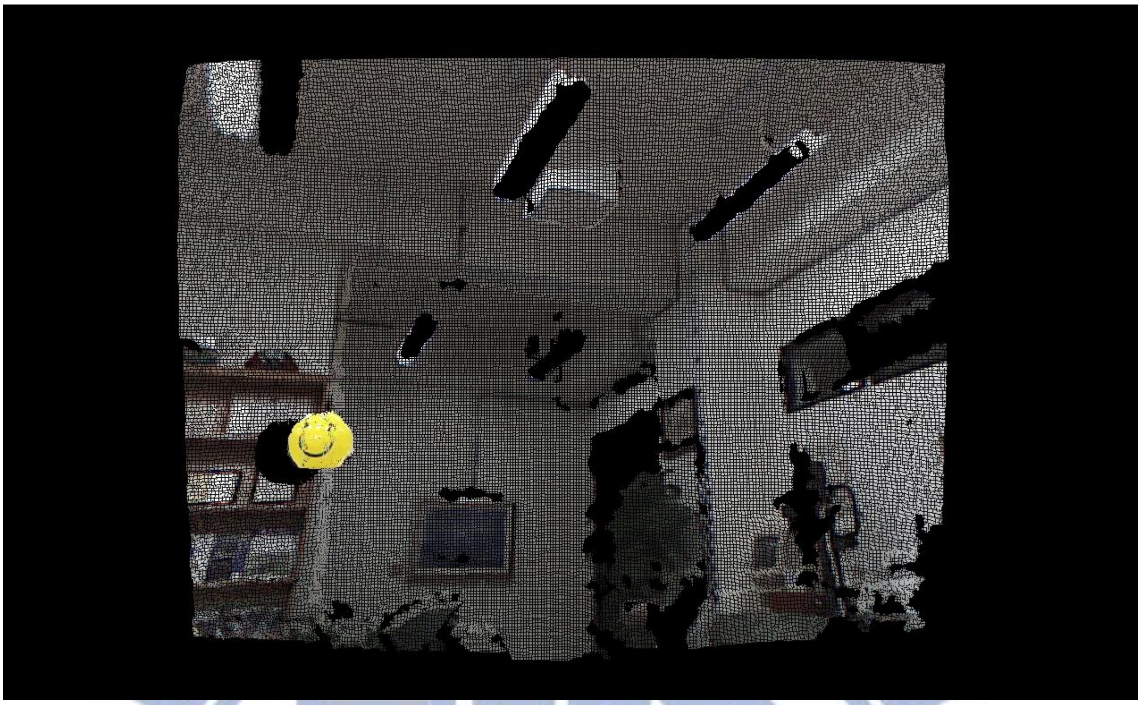


Figure 5.15 Watermarked 3D image and the position of the watermark is in the left side of the image.

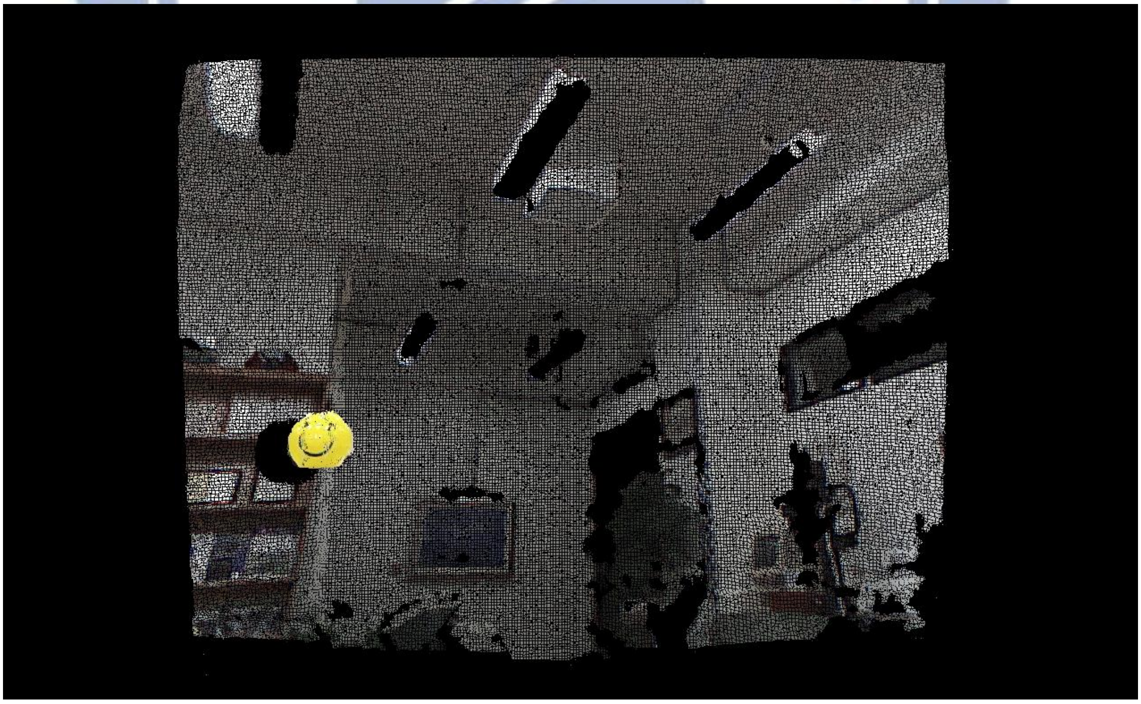
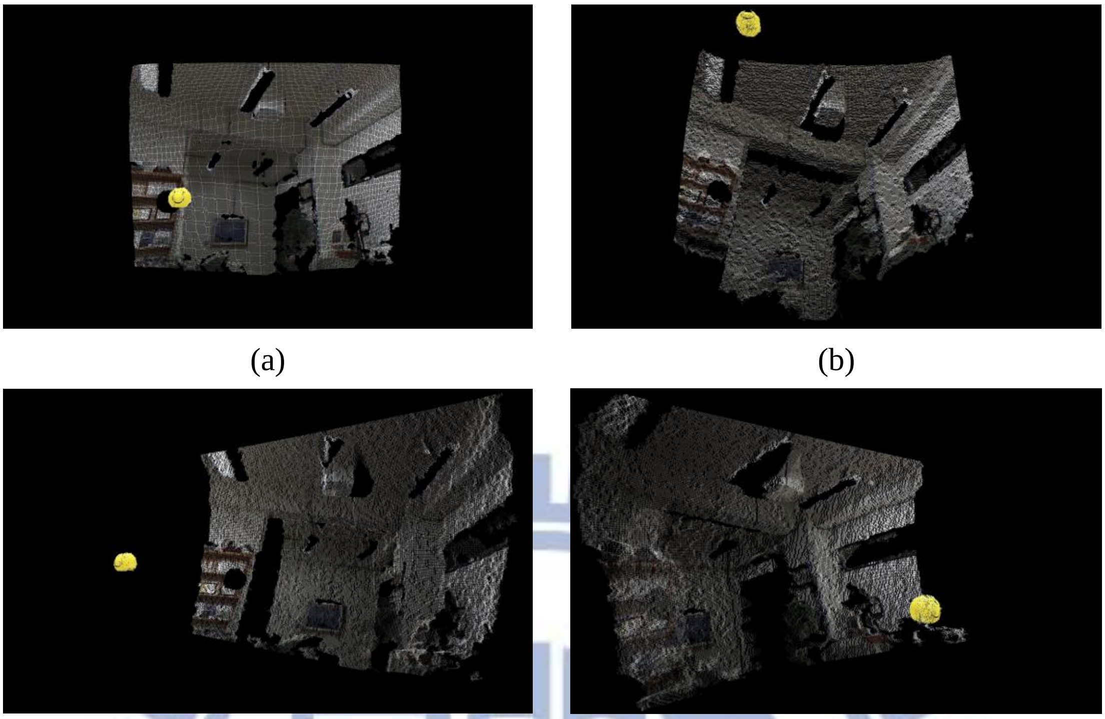


Figure 5.16 The watermarked 3D image with the recovery information embedding by the reversible contrast mapping method.



(a) (b)

(c) (d)

Figure 5.17 The results of the watermarked 3D image with the recovery information is embedded after the extraction process in different viewpoints. (a) the view of the position in front of the 3D image. (b) the view of the lowwer position in front of the 3D image. (c) the view of right position of the 3D image. (d) the view of left position of the 3D image.

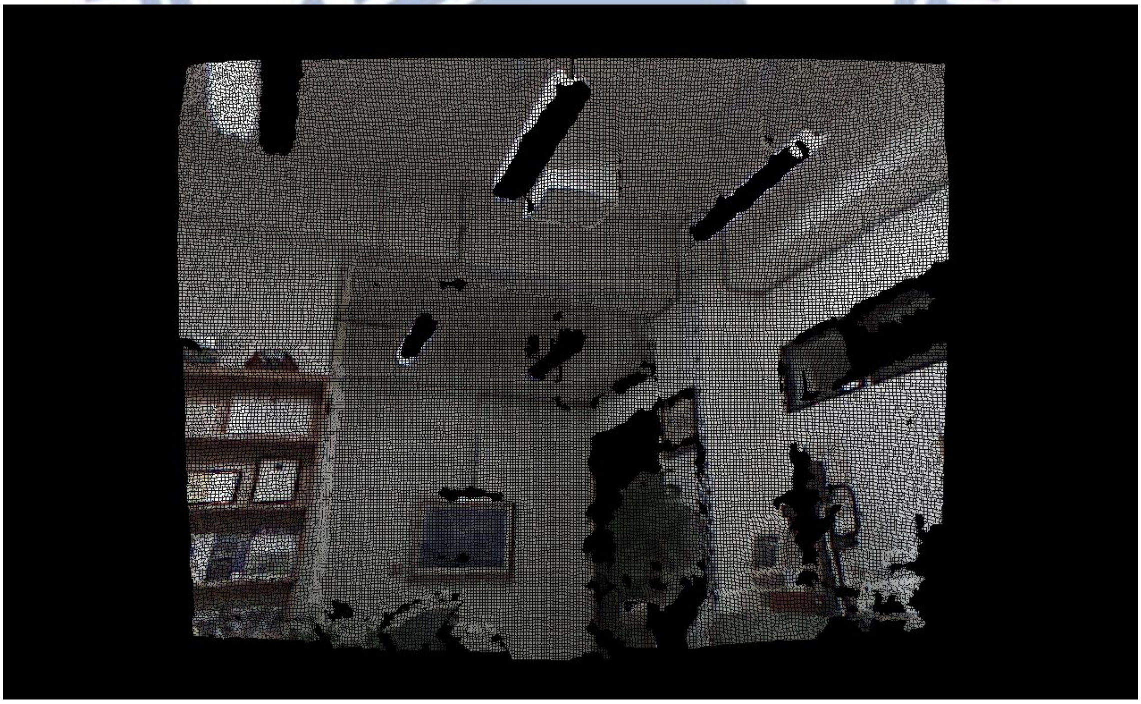


Figure 5.18 The result of eliminating the 3D watermark and recovering the covered region.

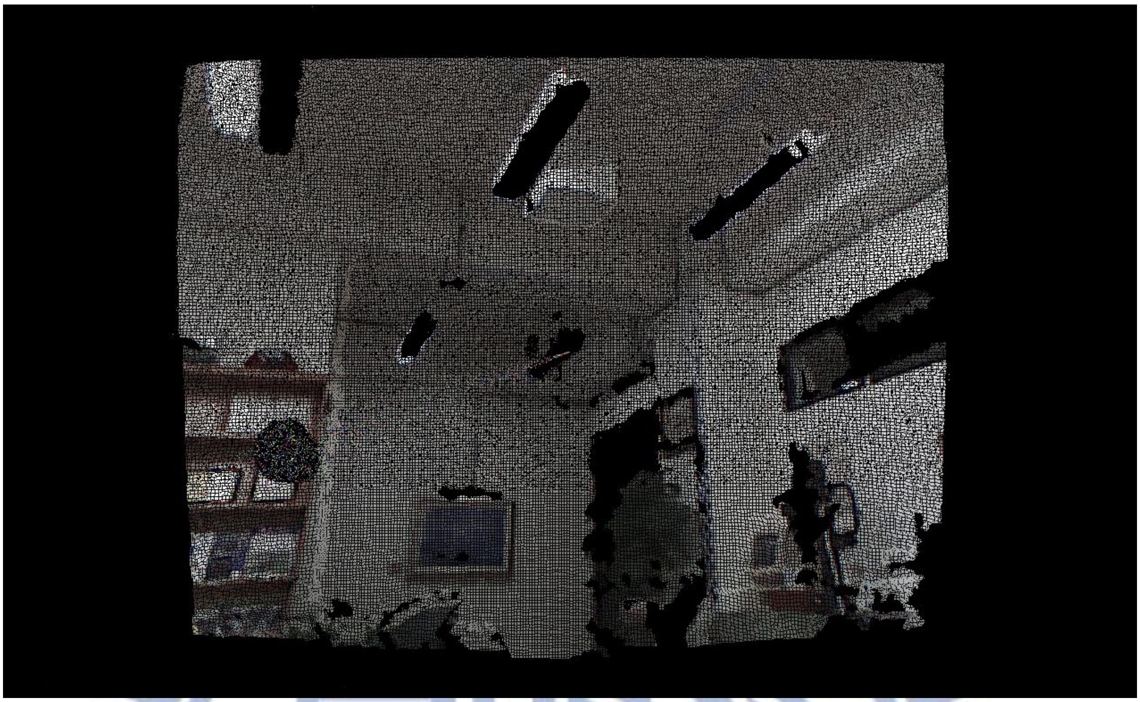


Figure 5.19 The result of eliminating the 3D watermark and recovering the covered region by using the wrong secret key to extract the hidden data.

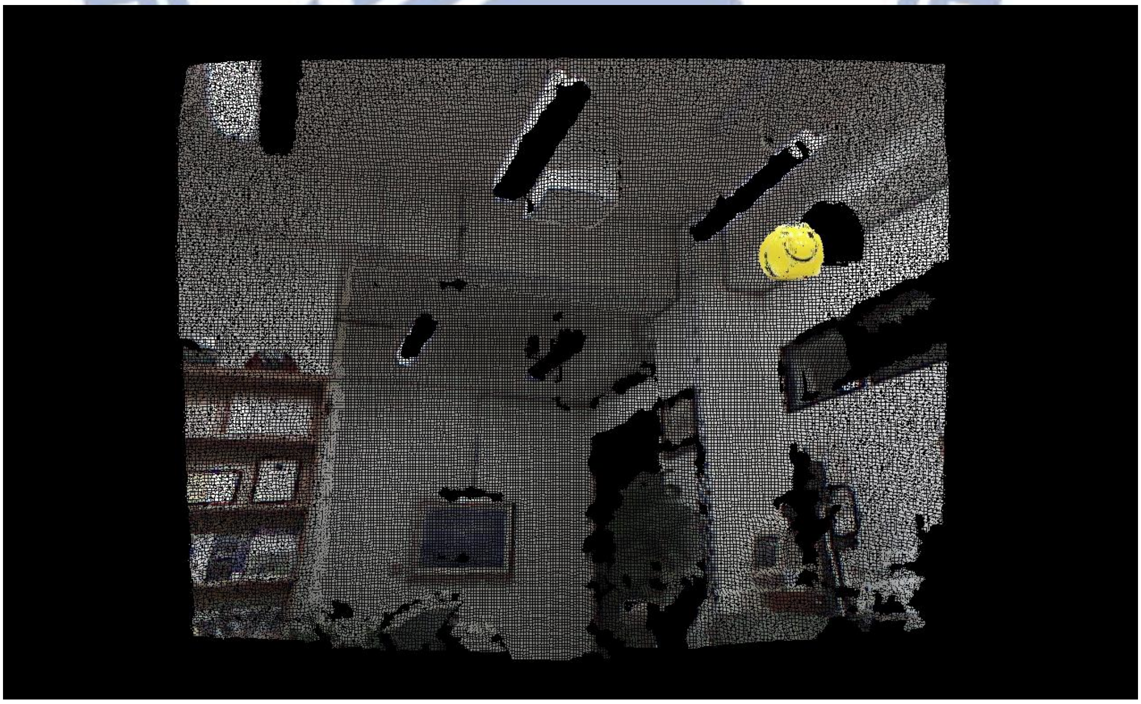


Figure 5.20 The watermarked 3D image with the recovery information embedding by the difference expansion method.

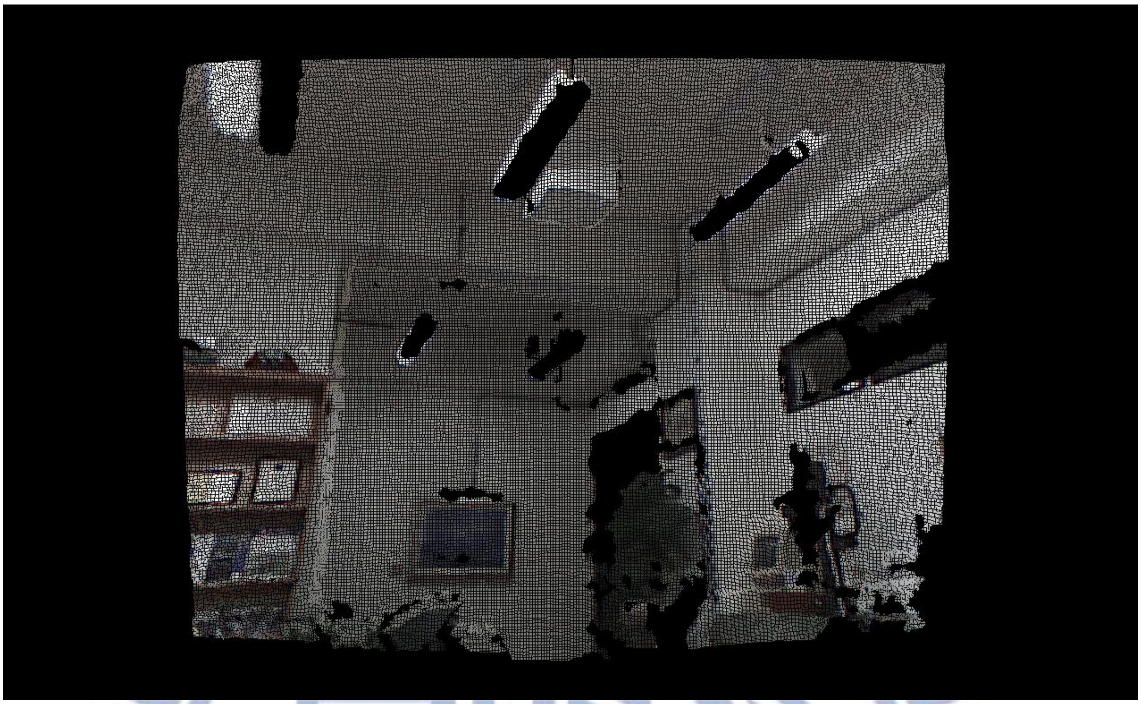


Figure 5.21 The result of eliminating the 3D watermark and recovering the covered region.

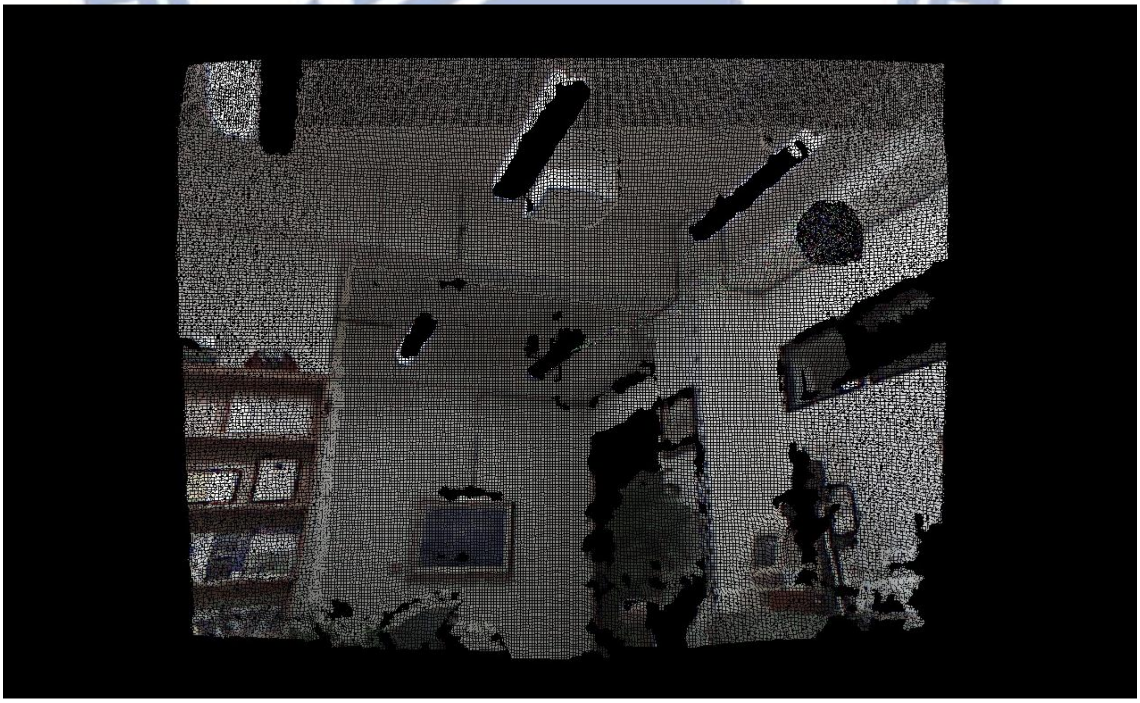


Figure 5.22 The result of eliminating the 3D watermark and recovering the covered region by using the wrong secret key to extract the hidden data.

5.6 Discussions and Summary

In this study, a method for copyright protection of KINECT images by 3D visible watermarking has been proposed. The proposed method embeds a 3D visible watermark into a 3D image, which results from combining the depth image and the color image acquired with a KINECT device, for protecting the copyright of the KINECT images. In addition, embedding the 3D visible watermark into the 3D image will cover some region of the original KINECT image. So we transform the information of the covered region into a data string *S* and hide it in the watermarked 3D image by the difference expansion method proposed by Tian [6] or by the reversible contrast mapping method proposed by Coltuc and Chassery [8], respectively. Besides, only authorized users who have the secret key can extract the recovery information from the watermarked 3D image. Then, the 3D watermark can be removed and the covered region can be recovered losslessly by using the recovery information. Consequently, only the authorized users can remove the 3D watermark losslessly.

Chapter 6
Conclusions and Suggestions for Future Works

6.1 Conclusions

In this study, realizing the importance and popularity of images acquired with KINECT devices, we have proposed several methods for data hiding via KINECT images and their applications, which includes authentication, covert communication, and copyright protection.

For authentication, a data hiding method for authentication of KINECT images by embedding authentication signals into the depth and color images acquired by the KINECT device is proposed. To achieve the goal of protecting the depth and the color image together, the proposed method utilizes the depth image to generate authentication signals and embedding them into the color image; and reversely uses the color image to generate authentication signals and embedding them into the depth image. Besides, the proposed method selects randomly the positions in the depth and color images to embed authentication signals in order to reduce the possibility for a malicious user to figure out the locations of the embedded authentication signals in the KINECT images. Furthermore, if the protected KINECT images are attacked, the proposed method can locate the pixels which are tampered with and repair some parts of the tampered regions.

For covert communication, a data hiding method for this purpose via KINECT images by interpolation at depth holes in proposed. To achieve the goals of enhancing the quality of depth images and hiding secret messages effectively, the proposed method utilizes a special type of feature, depth hole, in the depth image acquired by KINECT devices and proposes the use of a new interpolation technique. The secret message is transformed by using a secret key and the secret sharing scheme before the data hiding process is conducted.

For copyright protection, a method for protecting KINECT images by 3D visible watermarking is proposed. The proposed method embeds a 3D visible watermark into a 3D image to be protected, which results from combining the depth image and the color image acquired with a KINECT device. Because embedding the 3D visible watermark will cover some regions of the original 3D image, the original data of the regions are transformed into a recovery data sequence, which then is embedded in the watermarked 3D image by two schemes, one being the difference expansion scheme and the other the reversible contrast mapping scheme. For security enhancement, the order of the bits in the recovery data sequence is randomized by a secret key before the data embedding process is started, and the embedding locations in the 3D image are also selected randomly with the secret key such that only authorized users who have the secret key can extract the recovery information from the watermarked 3D image. Finally, the 3D watermark can be removed and the covered region can be recovered losslessly by using the recovery information, and again only authorized users can remove the 3D watermark.

Experimental results of the proposed methods have also been presented, which show the feasibility of the proposed methods for the applications of authentication, covert communication, and copyright protection.

6.2 Suggestions for Future Works

According to our experience in this study, several suggestions for future works are listed in the following.

1. In the proposed authentication method, the way to repair the tampered pixel can be modified to enhance the overall capacity of the repair process.
2. It is worthwhile to extend the proposed authentication method for KINECT images to fit other data types, such as surveillance videos acquired by KINECT devices or other similar equipments.
3. The way to find the depth holes in depth images can be modified to be more effective to increase the hiding capacity and enhance the quality of the processed depth image.
4. It is interesting to modify the proposed methods to fit the case of using multiple KINECT images for various applications.
5. It is beneficial to improve the data hiding method proposed in this study to reduce the distortion resulting from hiding data into KINECT images.

References

[1] Microsoft Kinect. Available online: http://www.xbox.com/zh-TW/kinect/ (accessed in May, 2013).

[2] S. H. Liu, H. X. Yao, W. Gao, and Y. L. Liu, “An image fragile watermark scheme based on chaotic image pattern and pixel-pairs,” *Appl. Math. Comput.*, vol. 185, no. 2, pp.869-882, 2007.

[3] C. W. Lee and W. H. Tsai, “A grayscale image authentication method with a pixel-level self-recovering capability against image tampering,” *Proc. 2011 IAPR Int’l Conf. on Machine Vision Applications*, Nara, Japan, pp. 328-331, June, 2011.

[4] C. W. Lee and W. H. Tsai, “Optimal Pixel-level Self-repairing Authentication Method for Grayscale Images under a Minimax Criterion of Distortion Reduction” *Optical Engineering*, 51 (5). 057006-1-057006-10, 2012.

[5] C. K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognition*, vol. 37, March 2004, pp. 469-474.

[6] J. Tian, “Reversible data embedding using a difference expansion,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, Aug. 2003, pp. 890-896.

[7] A. M. Alattar, “Reversible watermark using the difference expansion of a generalized integer transform,” *IEEE Transactions on Image Processing*, vol. 13, no. 8, Aug. 2004, pp. 1147-1156.

[8] D. Coltuc and J. M. Chassery, “Very fast watermarking by reversible contrast mapping,” *IEEE Signal Processing Letters*, vol. 14, no. 4, Apr. 2007, pp. 255-258.

[9] Y. Hu and B. Jeon, "Reversible visible watermarking and lossless recovery of original images," *IEEE Transactions on Circuit Systems and Video Technology*, vol. 16, no. 11, Nov. 2006, pp. 1423-1429.

[10] H. M. Tsai and L. W. Chang, "A high secure reversible visible watermarking scheme," *Proceedings of IEEE International Conference on Multimedia and Expo*, Beijing, China, July 2007, pp. 2106-2109.

[11] D. C. Wu and W. H. Tsai, "Spatial-domain image hiding using an image differencing," *IEEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 1, Feb. 2000, pp. 29-37.

[12] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, "A DCT domain visible watermarking technique for images," *Proceedings of IEEE International Conference on Multimedia and Expo*, New York, USA, vol. 2, Jul. 2000, pp. 1029-1032.

[13] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.