國立交通大學

多媒體工程研究所

碩士論文

利用 3D KINECT 影像做視訊監控應用上之隱私保護與秘密隱藏

Privacy Protection and Secret Hiding via 3D KINECT Images for Video Surveillance Applications

研究生：曾頌賢

指導教授：蔡文祥　教授

中華民國一百零二年六月

利用立體 KINECT 影像做視訊監控應用上之隱私保護與秘密隱藏

Privacy Protection and Secret Hiding via 3D KINECT Images for Video Surveillance Applications

研 究 生：曾頌賢　　Student：Chung-Yin Tsang

指導教授：蔡文祥　　Advisor：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

利用立體 KINECT 影像做視訊監控應用上之隱私保護與秘密隱藏

研究生: 曾頌賢　　指導教授: 蔡文祥

國立交通大學多媒體工程研究所

摘要

隨著立體電腦視覺技術的快速發展，3D 影像設備越來越普及，可在各種條件下讀取 3D 多媒體資訊。微軟的 KINECT 設備即是一著名的例子，它不僅可以捕捉彩色影像，更可以捕捉深度資訊，讓各種應用中的 3D 資訊擷取變得更為方便。

本研究提出了三種方法來達到在 3D 視訊監控中保護隱私及進行 3D 影像偽裝的目的。第一種方法可在 3D 視訊監控中保護使用者所選擇的隱私區域。該法利用可逆的映射函式，將隱私區域的影像轉變為與背景影像相似的偽裝影像，藉以產生一受保護視訊。第二種方法將該法延伸，做到隱藏私密性動態活動的作用，並藉由加速穩健特徵(speeded up robust features, SURF)和匹配演算法(matching algorithm)來自動偵測動態區域，作為私密影像的部分。第三種方法運用可逆的對比映射（reversible contrast mapping, RCM）方法來偽裝 3D 影像，可以將指定的 3D 秘密影像轉變成 3D 偽裝影像，並可無失真地恢復原來 3D 秘密影像的本貌。根據 RCM 的方法可以把 3D 秘密影像隱藏在 3D 偽裝影像中，藉以產生資訊隱藏的效果。

本論文所提出的方法皆獲得良好的實驗結果，證明其在實際應用上可行。

Privacy Protection and Secret Hiding via 3D KINECT Images for Video Surveillance Applications

Student: Chung-Yin Tsang Advisor: Wen-Hsiang Tsai

Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

ABSTRACT

With the rapid development of stereo-vision technology, 3D imaging devices become more and more popular. Many types of such devices have been invented to acquire 3D multimedia information under various conditions. One famous example is the KINECT device manufactured by Microsoft. It can capture not only color images but also depth images, making it easier to get 3D information for uses in various applications. In this study, we propose three methods for video surveillance applications with the aims of privacy protection in 3D video surveillance as well as 3D image steganography.

The first method is proposed for protecting a user-selected privacy-sensitive region in a 3D surveillance video, in which reversible prediction-based mappings are used to convert the privacy-sensitive region in each image frame into a background image part, resulting in a protected video. The method is extended further to conceal private motion activities in surveillance videos, by which motion regions are detected automatically as privacy-sensitive image parts by the SURF extraction and matching algorithm. The reversible mapping scheme used in the two methods guarantees

lossless retrieval of the concealed privacy-sensitive image part from each image frame in the protected video.

Moreover, a method using the reversible contrast mapping (RCM) scheme for 3D image steganography is proposed, which can hide 3D secret images into 3D cover images, as well as recover the embedded 3D secret image losslessly. Specifically, the method hides 3D secret images into non-object holes in the 3D cover image according to the RCM scheme to yield a 3D stego-image which has a steganographic effect. Also embedded into the input 3D cover image is a recovery sequence which can be extracted from the 3D stego-image to retrieve the original 3D secret image.

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

ACKNOWLEDGEMENTS

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

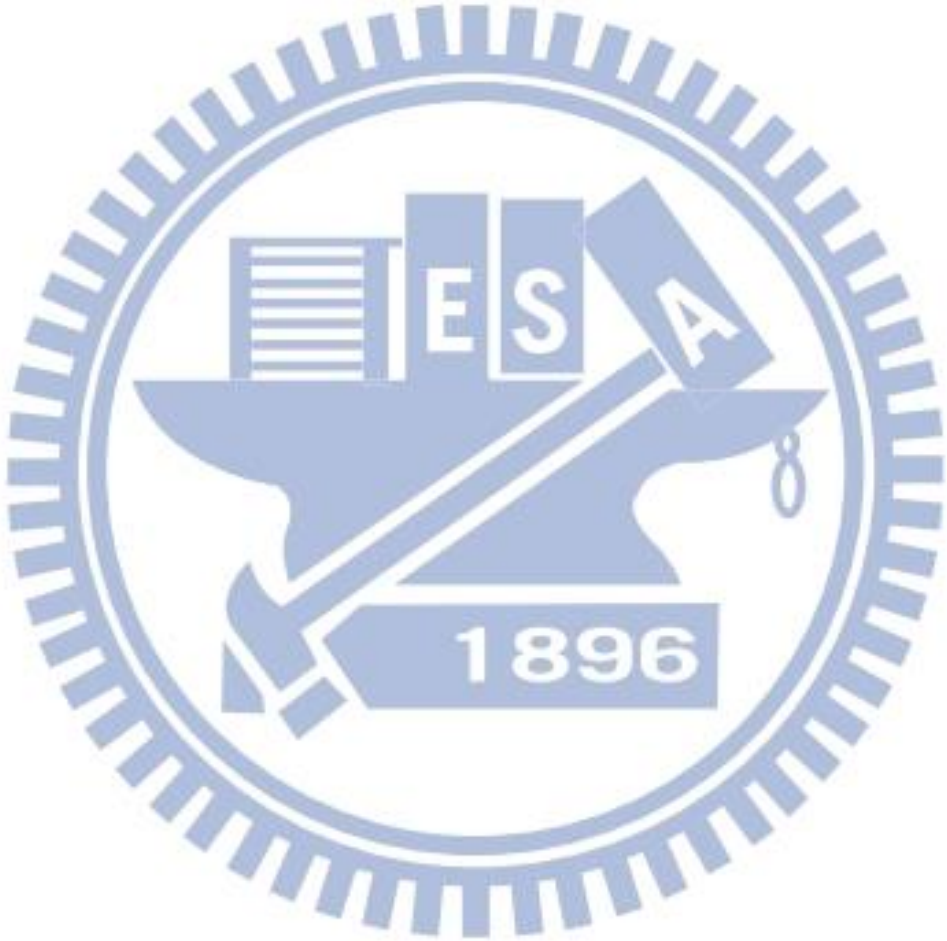
Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Institute of Computer Science and Engineering at National Chiao Tung University for their suggestions and help during her thesis study.

Finally, the author also extends her profound thanks to her dear family and boyfriend for their lasting love, care, and encouragement.

CONTENTS

LIST OF FIGURES

LIST OF TABLES



Chapter 1
Introduction

1.1 Background and Motivation

In recent years, the rapid development of the electronics technology and the widespread use of the Internet have created a rich life for people. In this trend, video surveillance is a common need in our environment, which is widely adopted in many applications, such as public space monitoring, casino security surveillance, special event recording, etc. Because people are concerned about many safety and privacy issues, the demand for video surveillance is increasing gradually. Consequently, more and more cameras for video surveillance are installed everywhere in our living environments. People think that video surveillance could bring us a more secure life, but they are also worried about the resulting infringement of their personal privacy.

More specifically, video surveillance becomes increasingly intrusive day by day with the increase of the number of deployed surveillance cameras in people's living environments. Also, the surveillance camera systems monitor public spaces, watch individuals' activities, and transform acquired images/videos into permanent data. Each of such systems is not just a tool used in some people's life, but also "keeps an eye" on the people all the time. In order to alleviate the concern of the public about invasions of privacy, it is necessary to develop new video surveillance techniques to protect the privacy of the people's life.

In addition, lots of data hiding techniques have been developed in the past decades. These techniques are widely used in quite a large number of applications,

such as covert communication, digital watermarking, multimedia authentication, and so on. However, the capability of the existing data hiding techniques for protecting privacy-sensitive data is still weak; it can only conduct the protection work by simple deleting or withholding of information in general.

Moreover, with the rapid development of stereo-vision technology, 3D imaging devices become more and more popular. Many types of such devices have been invented to get 3D multimedia information under various conditions. One famous example is the KINECT device manufactured by Microsoft. The KINECT device enables a user to control and interact with the Microsoft Xbox 360 through a natural user interface using gestures and spoken commands without the need to touch a game controller.

Specifically, the information of the depth image, in addition to that of the color image, acquired with the KINECT device may be utilized as an important hint to solve many 3D-related problems. For example, an object may not have consistency in color and texture, but it must occupy a solid region in space. Also, the color image taken by the KINECT device has advantages over a 2D intensity image. For example, the former is robust in the change of both color and illumination. Accordingly, in the application of video surveillance, we can get easily from 3D image data such information as an intruding person, his/her height and thickness, and so on. Because of these characteristics and usefulness of 3D image data, the issue of protecting 3D image data in various applications becomes more and more important. In this study, it is desired to propose methods for protecting privacy-sensitive in 3D images/videos.

Finally, with the popularity of 3D multimedia increasing day by day, it is in urgent need to develop data hiding techniques for 3D images. Specifically, it is desired to propose methods for protecting KINECT images (including the depth and color images) using data hiding techniques in order to prevent the information from

being stolen by hackers. Besides, due to the popularity of the Internet, image data are shared frequently on the Internet; therefore, it is also desired to propose methods for 3D steganography via KINECT images, by which a user can send secret data to other persons via the Internet or keep them securely in any digital storage. The details of these proposed methods will be described in subsequent chapters.

1.2 General Review of Related Works

The video surveillance becomes ubiquitous in our daily life. In this study, we try to use the information hiding techniques for privacy protection in the video surveillance. In previous studies, many video surveillance techniques have been developed for uses in different fields, such as the privacy protection in video surveillance [2-4], information hiding via images [5-7], and image steganography [8-10]. All these techniques will all be introduced in Sections 2.1 through 2.3.

As for the KINECT sensor, many researches using KINECT images have been published and various methods for human detection have also been proposed in the past few years [11-18]. Most of these researches are based on images that are taken by visible-light cameras, which are normally what human eyes "see." Uses of color images so acquired will encounter difficulties in perceiving the shapes of objects with articulated poses or when the background is cluttered. Instead, with the depth image taken with the KINECT device as aid, the mentioned problem may solved more easily. The details of the KINECT device and a review of the related techniques will be introduced in Section 2.4.

For the topic of motion detection which is useful detecting human activities in 3D images, many methods [19-24] have been developed to detect moving objects, such as temporal differencing, background subtraction, etc., and they will be reviewed

in Section 2.5.

1.3 Overview of Proposed Methods

In this study, we begin by reviewing some existing techniques for privacy protection in video surveillance. Then, we propose a method for protecting selected privacy-sensitive areas in the depth and color images acquired by the KINECT device. The main idea is easy to understand, just like the use of a "sticker" to cover a privacy-sensitive area in a video surveillance image. In order to implement this idea, a technique of prediction-based mapping is adopted [25], which involves both the color and depth images.

The prediction-based mapping technique can also be applied to other applications. Specifically, we apply it to protect privacy-sensitive motion activities in image sequences or videos in this study. In this method, the parts of the motion activities in a given image sequence or video are segmented out automatically. Moreover, we use information hiding techniques to embed the privacy-sensitive image parts into image sequences or videos. For this, admittedly it requires a huge data embedding capability to embed the privacy information by traditional information hiding techniques. Therefore, we use a reversible mapping function which allows the mapped values to be controllable in magnitudes, and synchronizes removals of corresponding areas in both color and depth images.

More specifically, the first method proposed in this study embeds, in a way of meaningful disguise, a specific privacy area in surveillance images/videos against a pre-selected background image part, using not only the color image but also the depth image in the embedding process. The results can be shown in 3D ways. And in the second method, the first method is extended to protect the privacy-sensitive motion

activities by detecting human activities in images at first and regarding the detected parts as mobile privacy areas.

Finally, considering the daily-increasing popularity of the 3D images which may be constructed from the use of the KINECT device and in order to reach the goal of hiding information in 3D images, we propose as well a method for 3D steganography via KINECT images utilizing 3D image processing techniques.

The detailed descriptions of the above-mentioned proposed methods will be presented in the subsequent chapters.

1.3.1 Definitions of terminologies

The definitions of some related terms used in this study are introduced as follows.

1. Privacy-sensitive image: a privacy-sensitive image is an image which includes privacy-sensitive contents and needs to be concealed.
2. Background image: a background image is a portion of an image used to cover part of a privacy-sensitive image.
3. Camouflage image: a camouflage image is an image produced by disguising a privacy-sensitive image to make it similar to a background image.
4. Protected image: a protected image is a stego-image produced by embedding some recovery information into a camouflage image.
5. Recovery sequence: a recovery sequence is a sequence which records the location of the privacy-sensitive image and the removed bits.
6. Recovered image: a recovered image is an image produced by removing embedded data from a stego-image.
7. Recovery process: a recovery process recovers the original cover image from a stego-image.

8. Motion region: a motion region is an area containing motion objects in an input video, which are obtained from a motion detection process.
9. Secret image: a secret image is an important image that should be protected properly and not be revealed to unauthorized people.
10. Target image: a target image is an image which is provided by the user and used to produce a camouflage image.
11. 3D image: a 3D image is one constructed from combining the depth and color images acquired with a KINECT device.

1.3.2 Brief description of proposed methods

(A) Protection of privacy-sensitive regions in surveillance videos acquired by a KINECT device —

A method for protection of privacy-sensitive regions in surveillance videos acquired by the KINECT device is proposed in this study. This method aims to protect privacy-sensitive images by using a reversible one-to-one prediction-based mapping function proposed by Liu and Tsai [1].

First, we generate a prediction-residue image with small pixel values. Then, we map a privacy-sensitive image and a pre-selected background image together into a third image, called a camouflage image, through the use of the above-mentioned function proposed by Liu and Tsai [1]. The resulting camouflage image is similar to the selected background image and it is hard to tell their differences by human eyes.

At last, in order to recover the privacy-sensitive image, we embed the start and end positions of the selected privacy-sensitive region into the camouflage image in order to generate a new camouflage image involving both the color and depth images,

called the protected image. The details of these processes will be described in Chapter 3.

(B) Protection of privacy-sensitive motion activities in surveillance videos acquired by the KINECT device —

A method for protection of privacy-sensitive motion activities in surveillance videos acquired by the KINECT device is also proposed in this study. How to protect selected privacy-sensitive regions has already been studied in the last method, so for this method we propose the use of the speeded up robust features (SURFs) for detecting privacy-sensitive motion activities of specific people in given image sequences or videos.

At first, we detect privacy-sensitive motion events in image frames, and then segment out the region enclosing each motion event in the frame, called protected region. In addition, we modify the appearance of the background image in the protected region. After these steps, we integrate the resulting background image and the privacy-sensitive image together to create a camouflage image by using the previously-mentioned prediction-based mapping function.

At last, we embed the start and end positions of the protected region and the parameters into the camouflage image which involves the color and depth images. The details of the proposed method and the employed SURF extraction technique will be described in Chapter 4.

(C) 3D Steganography via KINECT Images —

A method for 3D steganography via KINECT images is proposed in this study. In this method, a technique is proposed to recover the secret image from the camouflage

image only by use of the embedded recovery information.

At first, we modify the pixel values of a given 3D cover image according to certain color and coordinate tables. Next, we embed the secret image into the cover image to produce a camouflage image. The original 3D image data we want to recover should be saved, and the recovery information for this purpose is created, called the recovery sequence. Then, we transform the resulting camouflage image into color and coordinate tables.

Finally, the recovery sequence is embedded into the resulting camouflage image by a LSB-modification scheme. With this recovery sequence, the modified cover image part during the data embedding process as well as the hidden secret image can both be retrieved losslessly. The details of these processes will be given in Chapter 5.

1.4 Contributions

Some major contributions achieved in this study are listed as follow.

(1) A method of data hiding using both prediction-based mapping and reversible contrast mapping are proposed.

(2) New applications of the proposed data hiding techniques in video surveillance and steganography are suggested.

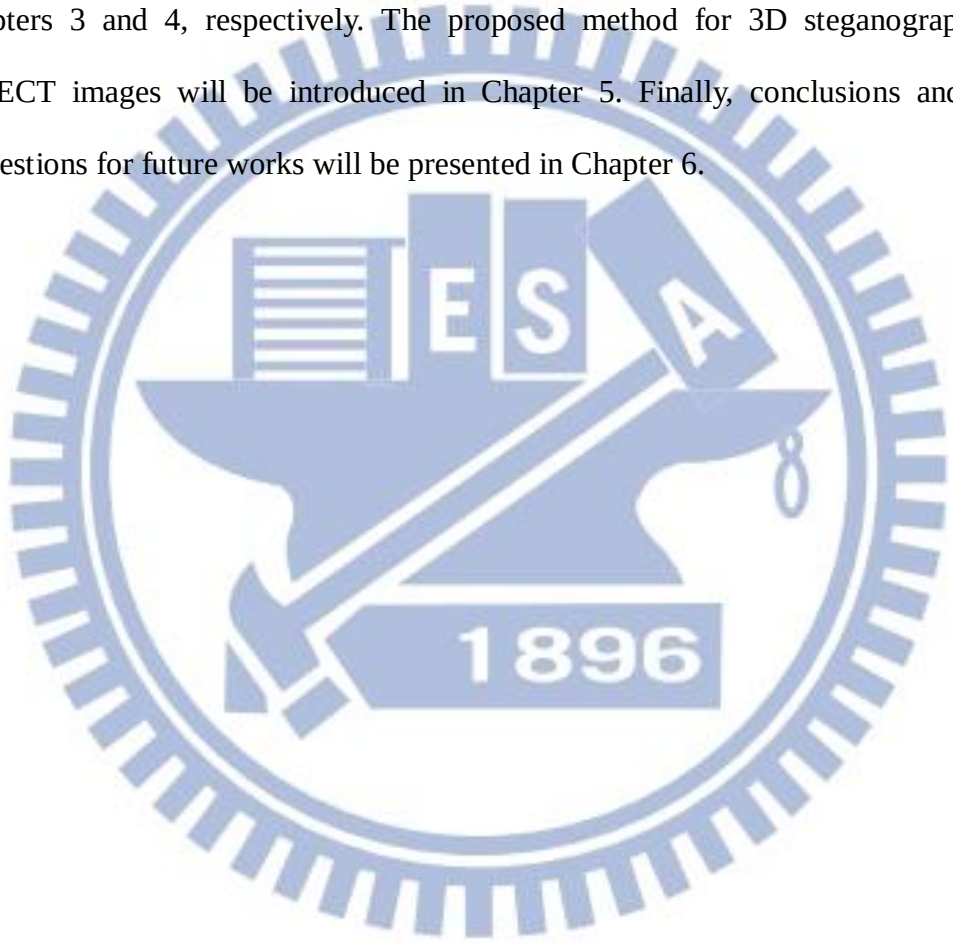
(3) A method is proposed to remove the privacy-sensitive image parts from given image sequences or videos, as well as to retrieve it when required.

(4) A method is proposed to synchronize removals of corresponding areas in both the color and depth images.

(5) A method for embedding a 3D image into another with a 3D steganography effect is proposed.

1.5 Thesis Organization

In the remaining parts of this thesis, the related works about privacy protection in video surveillance, data hiding in images and 3D steganography via KINECT images are reviewed in Chapter 2. The proposed methods for protecting privacy-sensitive regions and privacy-sensitive motion activities in surveillance videos are described in Chapters 3 and 4, respectively. The proposed method for 3D steganography via KINECT images will be introduced in Chapter 5. Finally, conclusions and some suggestions for future works will be presented in Chapter 6.



Chapter 2
Review of Related Works

2.1 Review of Techniques for Privacy Protection in Video Surveillance Applications

As global security concerns are now escalating, important video surveillance solutions have been proposed for applications of national security, law enforcement, public transportation, etc. They not only monitor people in various environments, but also expose unintentionally information with personal privacy. For this reason, privacy protection becomes indispensable in video surveillance. In this section, we will review those techniques proposed for privacy protection in video surveillance.

Paruchuri et al. [2] gave a survey of data hiding techniques for protecting privacy-sensitive contents in surveillance videos. To protect the privacy of selected individuals in videos, the usual way is to erase, blur, or re-render the image parts or frames of the individuals. Such modifications, however, destroy the authenticity of the original content of the surveillance video in concern. Paruchuri et al. [2] proposed a new rate-distortion based video data hiding algorithm for the purpose of storing the privacy-sensitive information in the compression domain. The algorithm embeds the privacy-sensitive information in optimal locations that minimize the resulting perceptual distortion and bandwidth expansion due to data embedding. Both reversible and irreversible embedding techniques were considered within the proposed framework, and extensive experiments were performed to demonstrate the

effectiveness of the techniques.

Elaine et al. [3] used the face recognition technique to automatically identify known people, such as against a database of driver-license photos. Moreover, they tracked people regardless of suspicion, guaranteeing that face recognition software will not recognize de-identified faces reliably, even though many facial characteristics were preserved. Also, the system obliterated relevant information, for example, object tracks or suspicious activities, from videos.

Dufaux et al. [4] also introduced a method to protect personal privacy by scrambling image regions containing personal information. As a result, the scene remained visible, but the privacy-sensitive information was not identifiable any more.

2.2 Review of Techniques for Information Hiding Techniques

With the advance of computer technology, information hiding has already become an indispensable part of our lives. In this field, videos, pictures, and digital audios are furnished with distinguishing abilities but imperceptible marks, which may contain a hidden patent or information, or even help to prevent modifications of its own.

Many data hiding methods have been proposed for still images. They can be classified into two main categories: (1) spatial-domain and (2) frequency-domain. In a spatial-domain method, the secret message is usually embedded by using LSB, statistical, feature, or block-based techniques. These techniques work with the pixel values directly, and images are generally manipulated by altering one or more bits of each byte of the image. On the other hand, in a frequency-domain method, a secret message is hidden in the coefficients of the image in the transform domain.

Generally speaking, the spatial-domain method is sensitive against attacks like image compression, resulting in quality degradations and content distortions, but the frequency-domain method is more robust. Therefore, data hiding in the frequency domain is less prone to attacks, but a lot of people prefer to use spatial-domain modification since it can hide more data.

For example, Ni, et al. [5] proposed a reversible data hiding algorithm for embedding data in the spatial domain by using the zero or the minimum point of the histogram and slightly modifying the pixel values. Also, Xuan et al. [6] proposed an approach to hiding data into the middle biplanes of the integer wavelet transform coefficients in the middle and high sub-bands of the frequency domain.

Besides, the most popular hiding technique may be the least-significant-bit (LSB) replacement. This method has been improved, extended, and revisited for years because the traditional LSB replacement method has many weaknesses, like distortion incurring, inherent fragility, etc. Even reversible LSB replacement methods havw been developed, which are very useful for visible watermarking and covert communication.

For example, Celik et al. [7] proposed a method for data embedding using lossless generalized LSB replacement. This method modifies the lowest levels, instead of the bit plane, of raw pixel values, and can recover the original image by compressing and transmitting the modified level values.

2.3 Review of Techniques for Image Steganography

Steganography is the art of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message. It

may be regarded as a form of security through obscurity.

Generally, messages will appear to be something else: images, articles, shopping lists, or some other cover texts. The advantage of steganography over cryptography alone is that messages do not attract attention to themselves. Plainly visible encrypted messages, no matter how unbreakable, will arouse suspicion and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

Lee et al. [8] proposed a method of watermarking which embeds an autostereogram into a cover image by the discrete cosine transform (DCT). This not only improves the low bit capacity but also the watermark efficiency because the random dots are distributed in the image space.

In [9], Wang and Chen presented an image steganography method that utilizes a two-way block-matching procedure in order to search for the highest similarity block for each block of the secret image. The indexes of these secret blocks are obtained in a block-matching procedure and recorded in the least significant bits of the cover image. Recently, Tsuda et al. [10] proposed a modified secure and high-capacity based steganography scheme for hiding a large-size secret image into a small-size cover image. The results show that the proposed algorithm for the modified steganography is highly secured in addition to having good perceptual invisibility.

2.4 Previous Studies on Applications of KINECT Devices

The KINECT device incorporates several types of advanced sensing hardware. The most notable is that it contains a depth sensor, a color camera, and a

four-microphone array, providing a full-body 3D motion capture device with face and voice recognition capabilities. The KINECT device is shown in Figures 2.1 and 2.2.

In more detail, the depth sensor in the KINECT device can emit an IR pattern, and capture the return signals as an IR image with a traditional CMOS camera that is fitted with an IR-pass filter.

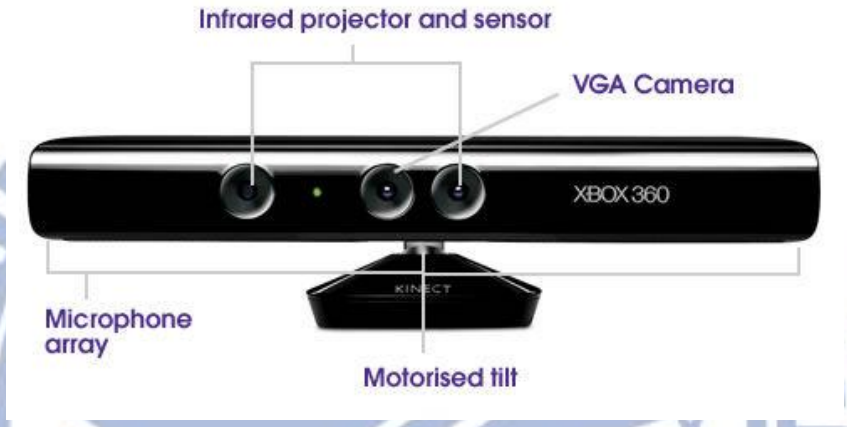


Figure 2.1 The outer appearance of a KINECT sensor.

Figure 2.2 The KINECT device is composed of an Infra-red (IR) projector, an IR camera, and an RGB camera

The image processor of the KINECT device uses the relative positions of the dots in the pattern to calculate the depth displacement at each pixel position in the image. It is noted that the actual depth values are the distances from the camera-laser plane rather than from the sensor itself. As such, the depth sensor can be seen as a device that returns the coordinates of 3D objects.

The hardware specifications of the KINECT device are described in many documents. The main ones are shown in Table 1.

Table 2.1Main hardware specifications of the KINECT device.

Property	**Value**
Angular Field-of-View	57 horz., 43 vert.
Frame rate	approx. 30 Hz
Nominal spatial range	640 x 480 (VGA)
Nominal spatial resolution (at 2m distance)	3 mm
Nominal depth range	0.8 m - 3.5 m
Nominal depth resolution (at 2m distance)	1 cm
Device connection type	USB (+ external power)

Three software frameworks are available for software developments using the KINECT device: Microsoft SDKs [11], OpenNI [12], and OpenKinect [13]. The Microsoft SDK is available only for Windows 7 whereas the other two frameworks are multi-platform and open-source software.

With the development of the depth camera technology, it is feasible to get high-quality color and depth images synchronously in realtime using the KINECT device now. Zhao et al. [16] compared the performances of different ways of

extracting interest points, and showed that the best performance could only be achieved by extracting interest points solely from the RGB channels, and then computing RGB-based descriptors and depth map-based descriptors together with those interest points.

In [17], Sung et al. proposed a method of using the KINECT device to extract human motions. The method was based on a hierarchical maximum-entropy Markov model (MEMM). From the RGBD images acquired with a Kinect sensor, the method extracts features and feeds them as input to a learning algorithm to train a two-layered Markov model which can capture different properties of human activities, including the correspondence between sub-activities and human skeletal features. The method considers a person's activity as composed of a set of sub-activities, and infers a two-layered graph structure from it by using a dynamic programming approach, as illustrated in Figure 2.3.

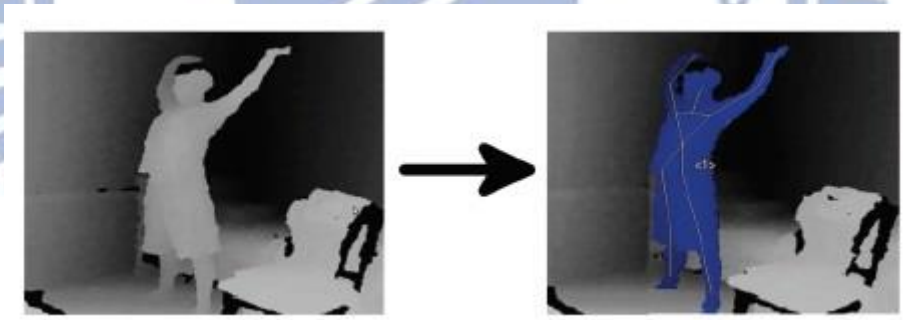


Figure 2.3 An example of extracting human activities from the RGBD data acquired with the Kinect sensor conducted by Sung et al. [17].

At last, Wang et al. [18] proposed a new feature descriptor, Pyramid Depth Self-Similarities (PDSS), for depth images. It was based on the idea that the depth information of people has high local self-similarities. Pedestrian detection was performed restrictively to single images, which involves three key aspects: feature, classifier, and detection strategy. To yield a better performance, it was suggested to

look for better features, as people present different somatotype and appearances, as illustrated in Figure 2.4.

Figure 2.4 An example of experimental results of feature extraction conducted by Wag et al. [18].

2.5 Review of Techniques for Motion Detection

Many motion detection techniques have been proposed to detect moving objects in videos [19-24], and some of them are reviewed in this section.

The main purpose of motion detection is to identify motion areas in a video, or to segment out motion objects from a video. A lot of content-based applications, such as smart signal processing, computer vision analysis, etc., have been developed. And their common works are often to detect motion objects firstly, and then to identify the properties of the objects for various applications.

Related techniques include human-face detection and recognition, motion-object tracking, content-based video retrieval, etc. Existing motion detection techniques can

be classified into two categories: one for use in the pixel domain [25-26] and the other in the compressed domain [27-29]. Generally speaking, the approaches used in the pixel domain have to fully decode a compressed video bitstream first, but they can be employed for videos coded according to different video coding standards. On the other hand, each of the approaches used in the compressed domain can perform a motion detection process by partially decoding a compressed video bitstream, but they can only be employed in videos coded according to specific standards.

Lipton et al. [19] proposed another approach which is based on temporal differencing in the pixel domain, where temporal differencing means the pixel-wise value differences between consecutive video frames. The basic idea of this approach is to compare video frames that are separated by a constant time in order to find moving objects. Haritaoglu et al. [20] also proposed a motion detection method which is based on background subtraction in the pixel domain. Both of the two methods built a statistical model for a background scene, and used the model to detect moving objects even though the background scene was not completely static.

Chapter 3
Protection of Privacy-sensitive Regions in Surveillance Videos Acquired by KINECT Device

3.1 Introduction

With the increasing public concern about personal privacy protection issues, it is desired to develop privacy protection methods for use in video surveillance systems. Besides, with the rapid development of stereo-vision technology, 3D imaging devices become more and more popular. Many types of such devices have been invented to get 3D multimedia information under various conditions. One famous example is the KINECT device manufactured by Microsoft.

Accordingly, in the application of video surveillance, we can get easily from 3D image data such information as intruding persons, their heights and thicknesses, and so on. Because of these characteristics and usefulness of 3D image data, the issue of protecting 3D image data in various applications becomes more and more important. In this chapter, we describe the proposed method for privacy protection of selected privacy-sensitive regions in image frames of surveillance videos taken with the KINECT device.

In Section 3.1.1, the related problem definitions are given. In Section 3.1.2, the related ideas are reviewed and the basic idea of the proposed method is described. The

principle behind the proposed method is based on the concept of merging of processed color and depth images to display 3D information, which we describe in Section 3.2. Detailed algorithms for privacy-sensitive region concealment and recovery based on the principle are presented in Section 3.3. Finally, some experimental results showing the feasibility of the proposed method are given in Section 3.4.

3.1.1 Problem Definition

As security surveillance is extensively applied in our living environment, there is a growing concern that the systems pose threats to personal privacy. Since the feeling of privacy is highly subjective and varying across cultures and individuals, the method of privacy protection should be adapted as much as possible to suit individual requirements. With regard to this demand, in the proposed method we allow an authorized user to specify a privacy-sensitive region R in a surveillance video in advance. The image content in R is defined as a privacy-sensitive image part which is not to be revealed to unauthorized people. The goal of the proposed method is to disguise the pre-selected privacy-sensitive image part as a corresponding background image part to conceal privacy-sensitive information in the image frames of the surveillance video. In addition, it is hoped that the protected image frames can be restored to include the original privacy-sensitive image part if a secret key is given as input.

Using traditional data hiding techniques to hide the privacy-sensitive image part may achieve this goal, but such techniques usually are time-consuming and demand large spaces for data embedding. Therefore, we design alternatively a general method for concealing the privacy-sensitive image part imperceptibly and recovering the original content of this image part from the resulting protected image losslessly. In

addition, even if a person knows the algorithms implementing the method, he/she still cannot retrieve the privacy-sensitive image part without the secret key. The security of the protected privacy in the surveillance video is thus ensured.

On the other hand, the KINECT sensor can be used to acquire the *depth* information, so the data can be used more extensively in applications than the 2D data. This kind of information is very important, so its correctness must be guaranteed. It is noted that the range of the depth values that are provided by the KINECT device is different from that of the general values of color-image. On the other hand, together with a depth image, a color image is taken simultaneously by the KINECT device. So, the depth image and the color image taken by the KINECT device at an identical instant of time should be protected together to keep their relation in time.

3.1.2 Review of Ideas of a Previous Study

Camouflaging is a method of hiding a secret. It allows a secret to be disguised by an object and so remain unnoticed. The major idea of the proposed method was inspired by the concepts of camouflaging and privacy protection in surveillance videos which were proposed by Lin and Tsai [25] as well as by the natures of the features of the KINECT images. The proposed method aims to produce a protected image by disguising a privacy-sensitive image part as a pre-selected background image part in a surveillance video, and to protect as a whole the color and depth images taken by a KINECT device at an identical instant. The proposed method utilizes the features of KINECT images and the range of pixel values in the depth image to achieve the goal of protecting the color and depth image together.

Specifically, the proposed method produces a camouflage image by using a prediction-based mapping, which a deterministic one-to-one (reversible) compound mapping function proposed originally by Liu and Tsai [1]. Following the function

proposed originally by Liu and Tsai [1], Lin and Tsai [25] integrated a new prediction technique into the prediction-based mapping to make the resulting camouflage image closely resemble the background image in appearance. The technique they proposed can estimate more effectively pixel values for use in the prediction-based mapping. Specifically, it uses the pixel-value similarity among adjacent pixels and employs a simple edge detection technique coming from the JPEG-LS standard [31].

3.2 Proposed Techniques of Synchronized Removals of Corresponding Areas in Color and Depth Images

In this section, we introduce the method proposed in this study which synchronously removes corresponding areas in color and depth images for privacy-sensitive image part concealment. The principle behind the proposed method is based on Lin and Tsai [25] as mentioned. In Section 3.2.1, the idea of the proposed method is described. And in Section 3.2.2, the detailed algorithms implementing the method are described.

3.2.1 Idea of Proposed Method

In the proposed method, at first a scheme is proposed to generate a new 3D image from the color and depth images acquired with a KINECT device. It is known that the KINECT device has a color camera and a laser sensor which are not aligned, leading to a displacement between the coordinates of the color image and those of the depth image. Therefore, we must correct this displacement to merge the two images to

produce a single *3D image*. We conduct this correction based on the use of a pinhole camera model.

The next major step of the proposed method is to disguise the privacy-sensitive image part as a pre-selected background image part to conceal the privacy-sensitive information in the image frames in the surveillance video. For this, we allow an authorized user to specify a privacy-sensitive region R in a surveillance video in advance, followed by the action of disguising the image content as mentioned above.

The third major step is to remove synchronously corresponding areas in the color and depth images in the mean time. The result is finally displayed in a 3D fashion. More details are described in the following.

3.2.2 Merging of Processed Color and Depth Images

To merge the depth and color images to produce a 3D image, a calibration of the camera parameters in advance is necessary. In this process, a rotation problem and a displacement problem in the 3D space should be solved at first. A solution to the rotation problem is to correct related parameters before mapping the KINECT images (including color and depth images) to produce an integrated 3D image. For this, some functions and parameters provided by the KINECT device and the Kinect-for-Windows SDK [11] can be utilized. In more detail, we tilt the field of view of each KINECT device to the zero-angle position using the tilt motor in the KINECT device before acquiring KINECT images; and to solve the displacement problem between the color image and the depth image, we use certain functions provided by the Kinect-for-Windows SDK [11].

After the above problems are solved, the original color and depth images are aligned in the same image coordinate system. But these 3D image data are just the 3D depth coordinates combined with the 2D image coordinates, so they must be

transformed into a single 3D space coordinate system integrally. For this purpose, we apply the principle of the pinhole camera model to conduct the transformation of the image coordinates into the 3D space coordinates.

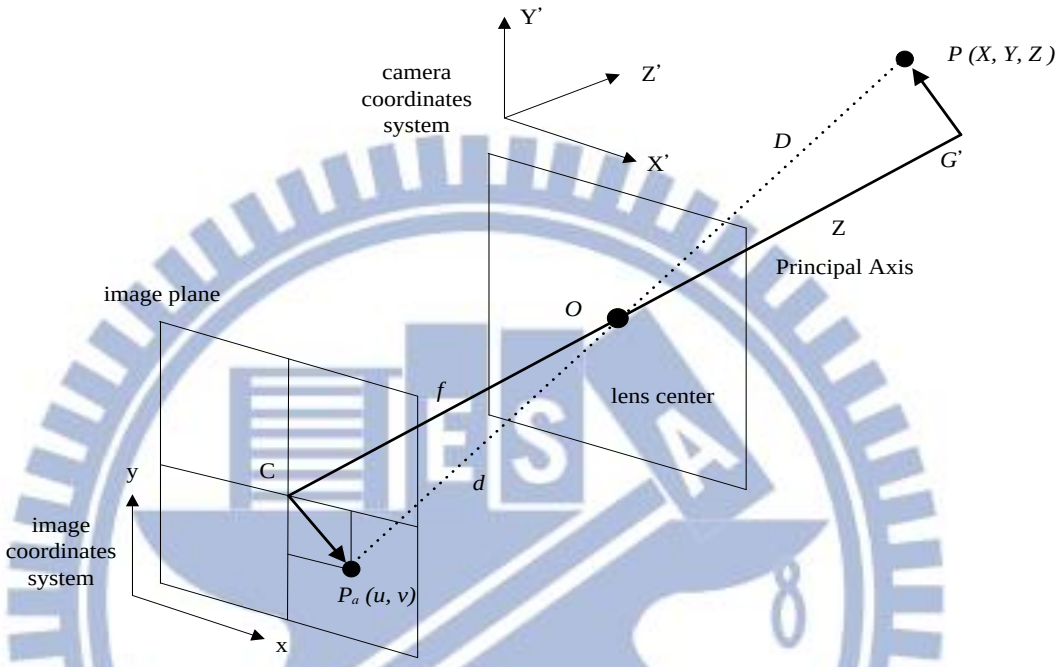


Figure 3.1 The pinhole camera model.

As illustrated in Figure 3.1, the pinhole model may be considered to be a simple camera with its center of projection (i.e., its lens center) located at *O* and its optical axis taken to be the *Z*-axis of the camera coordinate system. The focus point is on the image plane with a focal length *f*. A 3D space point $P = (X, Y, Z)$ in the camera coordinate system is projected onto an image point P_a on the image plane at image coordinates (u, v), where the image plane may be that of the depth image or the color image. The depth value *d* of the space point *P* is provided by the KINECT device, but we do not have its correct coordinates (X, Y, Z) in the 3D space (i.e., in the camera coordinate system). We can calculate them according to the similar-triangle principle.

In more detail, following the direction vector starting from the center *C* of the image plane, going through lens center *O* of the camera, and finally reaching at a 3D space point *G'* which is the projection of the space point *P* on the vector, we can see two similar triangles aside the direction vector. We can calculate the distance *d* between the image plane and the lens center O by the following equation:

$$d = \sqrt{u^2 + v^2 + f^2}\,. \tag{3.1}$$

Then, according to the principle of similar triangles, we can derive the following equalities:

$$\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y}. \tag{3.2}$$

which lead to:

$$u = \frac{fX}{Z};\ v = \frac{fY}{Z}.$$

Accordingly, we can derive the following equations to describe the relation between the image coordinates (u, v) and the corresponding coordinates (X, Y, Z) using homogeneous coordinates:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} u \\ v \end{pmatrix};\ \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \frac{1}{Z}\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.3}$$

Using the above results, we can start to merge the color and depth images acquired by the KINECT device. The detailed algorithm is described in Algorithm 3.1 below.

Algorithm 3.1: merging of color and depth images.

Input: a depth image *D*, a color image *C*.

Output: a 3D image *J* constructed from the color and depth images.

Steps:

Step. 1 Perform the process of Equations 3.1 through 3.3 to "calibrate" the pixels in color image C and those in depth image D, with a pixel C_p in the resulting C having values (C_r, C_g, C_b), and a depth pixel D_p in the resulting D having values (D_x, D_y, D_z).

Step. 2 Transform each pixel index C_i of color image C into the pixel index D_i of depth image D, where C is four times as big as D, by the function $D_i = \lfloor C_i/4 \rfloor$.

Step. 3 In a raster-scan order, take a pixel D_p from the depth image D and a corresponding pixel C_p from the color image C.

Step. 4 Knowing the extrinsic rotation R and translation T between the color and depth camera, express the mapping between color image pixel C_p and depth image pixel D_p by following equations: $\begin{pmatrix} C_r \\ C_g \\ C_b \end{pmatrix} = \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} R + T$.

Step. 5 Construct the 3D image J by $J_{xp} = -J_x \times \frac{f}{J_z} + u$; $J_{yp} = -J_y \times \frac{f}{J_z} + v$ to find the color information corresponding to each 3D point. The space coordinates (J_x, J_y, J_z) of a 3D point can be used to find the coordinates (J_{xp}, J_{yp}) of the corresponding image point, and draw these 3D point in the 3D space by the OpenGL.

In the above algorithm, the values of R and T are given by following equations:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}; R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$T = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ as extrinsic rotation R and translation T.

3.3 Proposed Method for Protecting Selected Privacy-sensitive Regions in Surveillance Videos

In this study, we adopt the processes for reversible prediction-based mapping from the previous study of Liu and Tsai [1] and Lin and Tsai [25]. About the processes for reversible prediction-based mapping, we will discuss problems encountered in applying the mapping, and propose solutions to them. Subsequently, they will be applied to security protection for video surveillance in Section 3.3.1. And the complete processes of privacy-sensitive region concealment and recovery are presented in Sections 3.3.2 and 3.3.3, respectively. The detailed algorithms about the proposed methods and the complete processes of concealing and recovering privacy-sensitive image parts will be presented in this section.

3.3.1 Review of application of reversible prediction-based mapping to privacy protection in surveillance videos

The method for privacy protection in surveillance videos is based on the use of the reversible prediction-based mapping proposed by Liu and Tsai [1], which is a deterministic one-to-one compound mapping of values. Besides, Lin and Tsai [25] proposed a principle of mapping and its use for protection of pre-selected privacy-sensitive regions will be described in this section.

As proposed in Liu and Tsai [1], a one-to-one mapping F_c with the function of $F_c(p) = p - c$ is created, where c is a parameter to be determined. As is well known,

the values of the adjacent pixels of P in the image are usually close to p because of contextual dependency. So, if we compute the average value a of them, a will be usually close to p and can be regarded as a *prediction* value of p. Therefore, we can take a as the parameter c of the function F_c above so that $F_c(p) = F_a(p) = p - c = p - a$ will be usually small because a is close to p in value. This function value will be called the *prediction residue* of p and denoted as r subsequently, i.e., $F_a(p) = p - a = r$.

Next, a second one-to-one mapping $F_b^{-1}(r) = r + b$ is performed, which adds r to the *target value* b, where F_b^{-1} is the inverse of F with parameter b. The resulting value is denoted as q. The overall 2-step mappings result in a *compound* one-to-one mapping function f with the following effect: $f(p) = F_b^{-1}(r) = F_b^{-1}(F_a(p)) = r + b = (p - a) + b = q$.

As stated above, $r = p - a$ is usually close to 0, so the value $q = f(p)$ will be close to the target value b, creating an effect of steganography. Therefore, we will call q a *stego-value*. Also, it is obvious that the smaller the prediction residue value r, the closer the stego-value q is to the target value b.

If it is desired to recover p from q, then the inverse f^{-1} of the compound one-to-one mapping function f can be used and the recovery is lossless, as can be seen from the following derivations:

$$f^{-1}(q) = [F_b^{-1}(F_a(q))]^{-1} = F_a^{-1}(F_b(q)) = F_a^{-1}(q - b)$$
$$= F_a^{-1}(p - a + b - b) = F_a^{-1}(p - a) = p - a + a = p.$$

Accordingly, to recover p from q, first we retrieve the prediction residue value $r = p - a$ by computing the value of the inverse F_b of the second mapping function F_b^{-1} with the stego-value q as input. This results in $F_b(q) = q - b = p - a + b - b = p - a = r$. Then, we use the same prediction scheme to compute the prediction value a from the values of the pixels adjacent to pixel P. Finally, we use the inverse F_a^{-1} of the first mapping function F_a to compute the original pixel value p of pixel P by $F_a^{-1}(r) = r + a$

$= p - a + a = p$.

It is noted that in the above recovery process, we have to compute the same prediction value a first, and then use it to recover the original value of the source pixel *P*. Based on this principle, Lin and Tsai [25] did not use the original value of the pixel *P*, but use only the values of the pixels adjacent to *P*, to compute *a* for the purpose of producing the identical prediction value.

In the previous method, conversion of a source value into a stego-value by two simple mapping function $F_a(p) = p - a = r$ and $F_b^{-1}(r) = r + b = q$ will cause some problems. The computed stego-value *q* might exceed the range of valid pixel values of *a*, *b*, and *p*. Then, Lin and Tsai [25] proposed another way to solve this problem.

To solve this problem, Lin and Tsai [25] adopted another one-to-one mapping function F_c proposed in [1] with $c = a$ or *b* such that the compound mapping $q = F_b^{-1}(F_a(p))$ did not exhibit this wrap-around problem. Based on this new function F_c, we describe how the corresponding mappings $F_a(p)$ and $F_b^{-1}(r)$ work, respectively, by Algorithms 3.2 and 3.3 below.

Algorithm 3.2: computing the value of a new mapping function $F_a(p)$ which does not cause the wrap-around problem.

Input: a prediction *a* and a source pixel value *p*.

Output: the prediction residue *r* of the mapping function $F_a(p)$ without causing the wrap-around problem.

Steps:

Step. 1 Initialize *r* to be zero.

Step. 2 Create a set *S* with 256 initial elements 0 through 255.

Step. 3 Find a value p' in *S* such that $|a - p'|$ is the minimum, preferring a smaller p' in case of ties occur.

Step. 4 If p' is not equal to p, then remove p' from S, increment r by one, and go to Step 3; otherwise, take the final r as the output.

Algorithm 3.3: computing the value of the inverse $F_b^{-1}(r)$ of the mapping $F_a(p)$ described by Algorithm 3.2.

Input: a target value b and a prediction residue value r.

Output: an stego-value q of the inverse mapping function $F_b^{-1}(r)$ of $F_a(p)$

Steps:

Step 1 Create a set S with 256 initial elements 0 through 255.

Step 2 Find a value q in S such that $|b - q|$ is the minimum, preferring a smaller q in case of ties.

Step 3 If r is larger than 0, then remove q from S, decrement r by 1, and go to Step 2; otherwise, take the final q as the output.

Based on the above two algorithms, the ideas for converting the source value p into a stego-value q and recovering p from q losslessly now can be described integrally by Algorithms 3.4 and 3.5, respectively, below.

Algorithm 3.4: converting a source pixel value to a stego-value which is close to a target pixel value.

Input: a source value p, a target value b, and the mapping F_c and its inverse F_c^{-1} described by Algorithms 3.2 and 3.3, respectively, where c is a parameter.

Output: a stego-value q.

Steps:

Step 1. Compute a prediction value a by a prediction technique.

Step 2. Perform Algorithm 3.2 to compute the prediction residue value $r = F_a(p)$.

Step 3. Perform Algorithm 3.3 to compute the output stego-value $q = F_b^{-1}(r)$.

Algorithm 3.5: recovering a source pixel value from a stego-value.

Input: the stego-value q produced by Algorithm 3.4 and the target b used there.

Output: the source value p.

Steps:

Step 1. Compute the prediction value a by the same technique as used in Step 1 of Algorithm 3.4.

Step 2. Regard the input values b and q as a and p, respectively, and take them as inputs to Algorithm 3.2 to compute a prediction residue value r.

Step 3. Regard a obtained in Step 1 as b, and take it and the value r obtained in Step 2 as inputs to Algorithm 3.3 to compute the source value p as the output.

The principle of reversible prediction-based mapping described above was applied to accomplish privacy protection in a surveillance video in the previous review. Lin and Tsai [25] regarded the source value p as a pixel value in a privacy-sensitive image part I_s to be protected, which appears within a region R, called privacy-sensitive region, in every surveillance video frame V other than a pre-selected video frame V_o taken as a background image.

Also, Lin and Tsai [25] used a background image part I_b in V_o, which corresponds to I_s in position in R but without privacy information, to replace I_s using reversible prediction-based mapping described above. Each pixel value in I_b is just a target value b mentioned previously. Then, as illustrated in Figure 3.2, we may use Algorithm 3.4 to find a prediction value a for each pixel in I_s by Step 1 of the algorithm, yield a prediction-residue image part I_r consisting of the prediction residue values r in region R by Step 2, and finally generate a camouflage image V' with stego-values q in region R by Step 3.

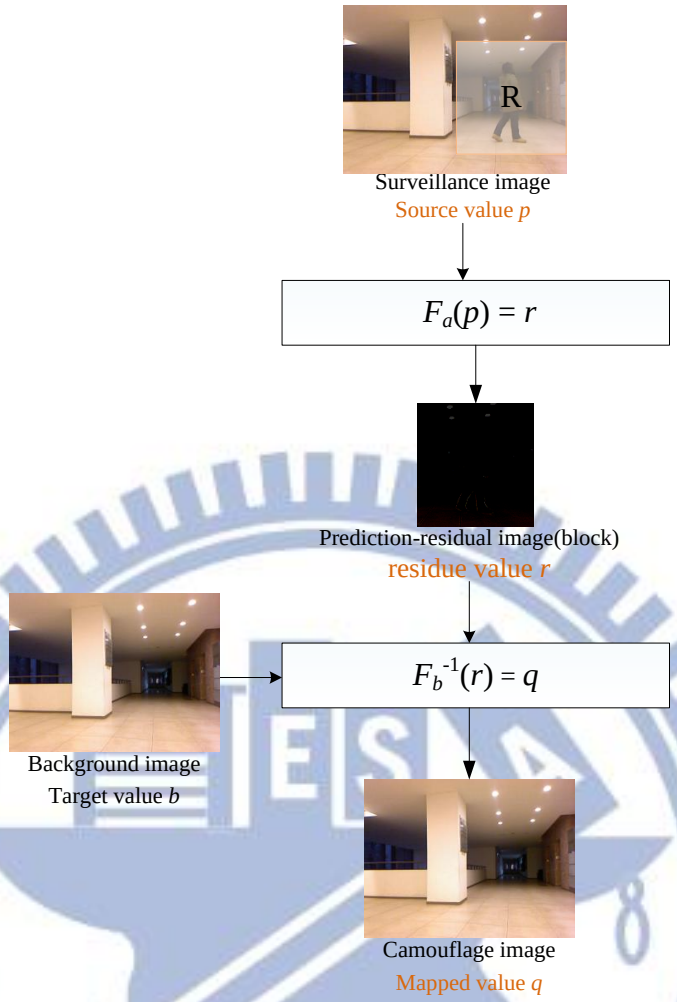


Figure 3.2 Illustration of concealment of a privacy-sensitive image part.

Also, the recovery of the original content of the concealed privacy-sensitive image part I_s may be carried out by Algorithm 3.5, as illustrated in Figure 3.3, by taking the camouflage image V' and the background image part I_b as inputs. The algorithm computes the same prediction value a for each pixel in region R by Step 1, yields the prediction-residue image part I_r by Step 2, and recovers finally the original privacy-sensitive image part I_s with source values p in R by Step 3, which may used to compose the original surveillance image V.

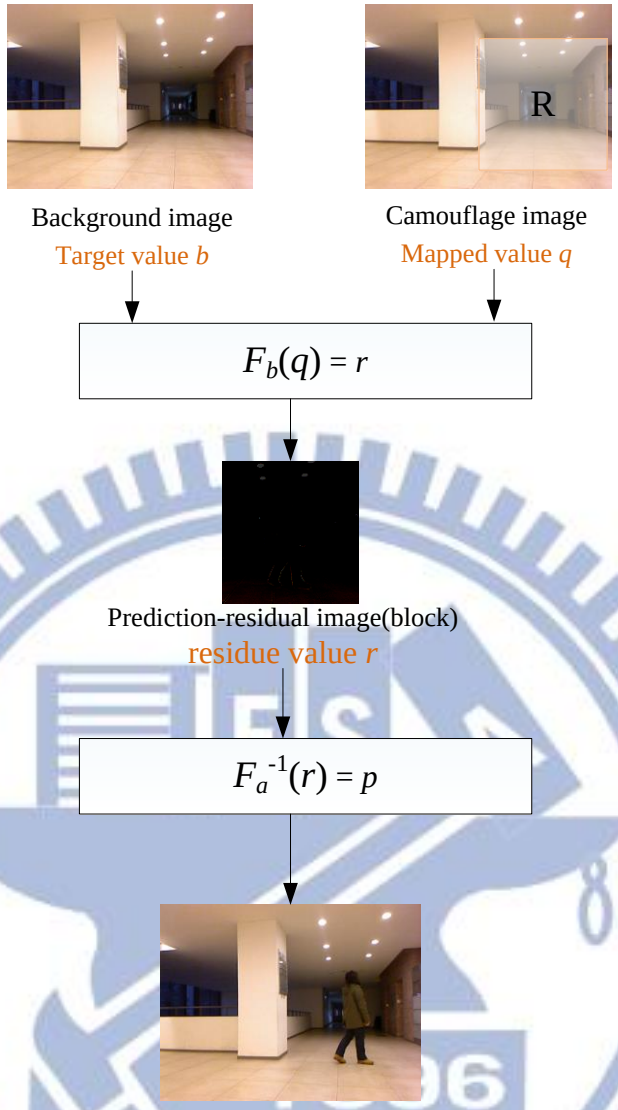


Figure 3.3 Illustration of proposed method of recovery of the privacy-sensitive image part from a camouflage image in a surveillance video.

An important principle to develop the function to compute the prediction value for the source value of each pixel in the privacy-sensitive image part I_s is that the function should use as input *only* the values of those pixels adjacent to the current pixel *P*, *excluding* that of *P* itself. The reason for adopting this principle is that we want to use the *same* prediction values for the prediction-based mappings to recover the original pixel values, and this principle ensures that the same prediction value can be computed for each pixel in I_s in *both* the privacy concealment and recovery

processes.

In this study, the proposed method for concealing pre-selected privacy-sensitive regions based on the concept of reversible prediction-based mapping and KINECT images. The proposed method about reversible prediction-based mapping and the KINECT images as described in the next section.

(A) Proposed content-based prediction method

In the JPEG-LS standard [31], a simple edge detection technique is used to determine the prediction value of each pixel. Furthermore, it is clear that each surveillance video frame exhibits strong spatial dependencies among its pixel values. It is useful to use such spatial dependencies to compute the prediction values for the source pixels in I_s. The median-edge-detection (MED) predictor adopted in the JPEG-LS standard [31] is based on this concept. Lin and Tsai [25] extended the MED predictor to design a new scheme for computing the prediction values. Let the value of the current pixel P be denoted as x and those of three neighbors of P as a, b, and c as illustrated in the *prediction template* of the JPEG-LS standard shown by Figure 3.4. The MED predictor is based on the concept of edge detection in the template, and computes the prediction value x' for x by Equation (3.2) below:

$$x' = \begin{cases} \min(a,b) & \text{if } c \geq \max(a,b); \\ \max(a,b) & \text{if } c \leq \min(a,b); \\ a+b-c & \text{otherwise.} \end{cases} \tag{3.2}$$

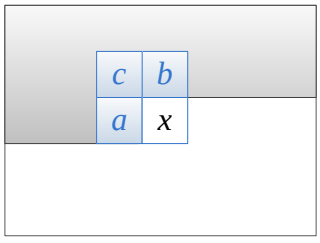


Figure 3.4 The prediction template used in the JPEG-LS standard.

The prediction scheme proposed by Lin and Tsai [25] used more pixels to compute the prediction value x' so that the prediction result can be more effective. In more detail, the following three steps are adopted in the proposed scheme: 1) using the values of some pixels in the 7×7 neighborhood of P as shown in Figure 3.5 to derive three candidate prediction values, denoted as E_h, E_v, and E_d, respectively, for the horizontal, vertical, and diagonal directions; 2) regarding E_h, E_v, and E_d respectively as the values of a, b, c in the prediction template as illustrated in Figure 3.4; and 3) computing the prediction value x' for x according to Equation (3.2) above.

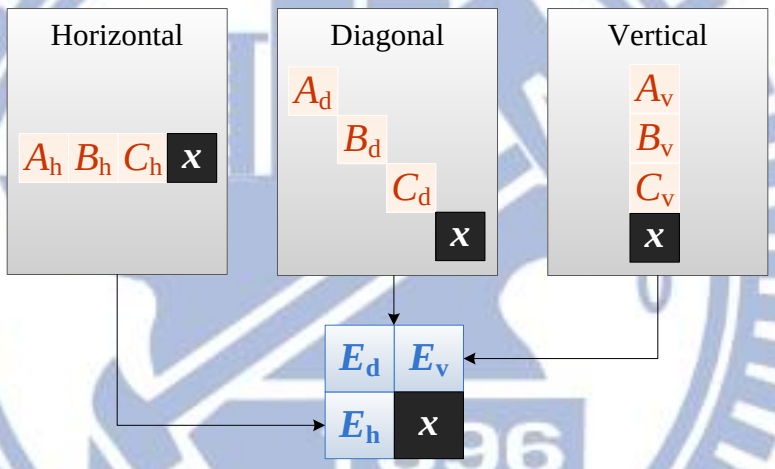


Figure 3.5 Prediction template used to derive candidate prediction values for a pixel P with value x.

Furthermore, to ensure that the same prediction value can be computed for each pixel P in I_s in *both* the privacy concealment and recovery processes, Lin and Tsai [25] kept track of the pixels that have been processed throughout the concealment process. Also, the pixels outside the privacy-sensitive region R, whose values need not be predicted, are regarded as having been processed.

(B) Quality improvement in the mapping results

At last, Lin and Tsai [25] applied the proposed prediction method and the

mapping table to map a privacy-sensitive image part I_p (part of a surveillance image) into a prediction-residue image I_r, and then to map the prediction-residue image I_r and a background image part I_b together into a camouflage image I_c by the inverse-mapping table. As stated earlier, the smaller the prediction residue values in I_p is, the closer the stego-values in I_c is to the target values in I_b. It can be seen that the quality of the generated camouflage image is contingent on the accuracy of the prediction method.

The proposed scheme cannot predict accurately for edge pixels in I_s since the value of an edge pixel might change dramatically with respect to the those of its adjacent pixels. For this reason, the prediction residue values of the pixels on the edge area will be larger than those not on the edge area. This weakness of the above prediction scheme will cause the yielded camouflage image to reveal the contours of prominent objects existing in I_s. An example of the results with such a phenomenon is shown in Figure. 3.6(c) with Figures. 3.6(a) and 3.6(b) as the privacy-sensitive image part I_s and the background image part I_b, respectively (enclosed by the rectangles). A remedy scheme is proposed to solve this problem, which includes the following two major ideas.

Stage 1 – reducing the prediction residue value

As mentioned previously, larger residue values produce distortions at edge areas in the resulting camouflage image. Accordingly, it seems feasible to solve this problem by reducing the residue values at the edge areas in the prediction-residue image I_r. However, this requires recording of the edge pixels' positions where such value reductions have been conducted for the purpose of recovering the original privacy-sensitive image part I_s later. Such a recording work will demand a large storage space and is so impractical.

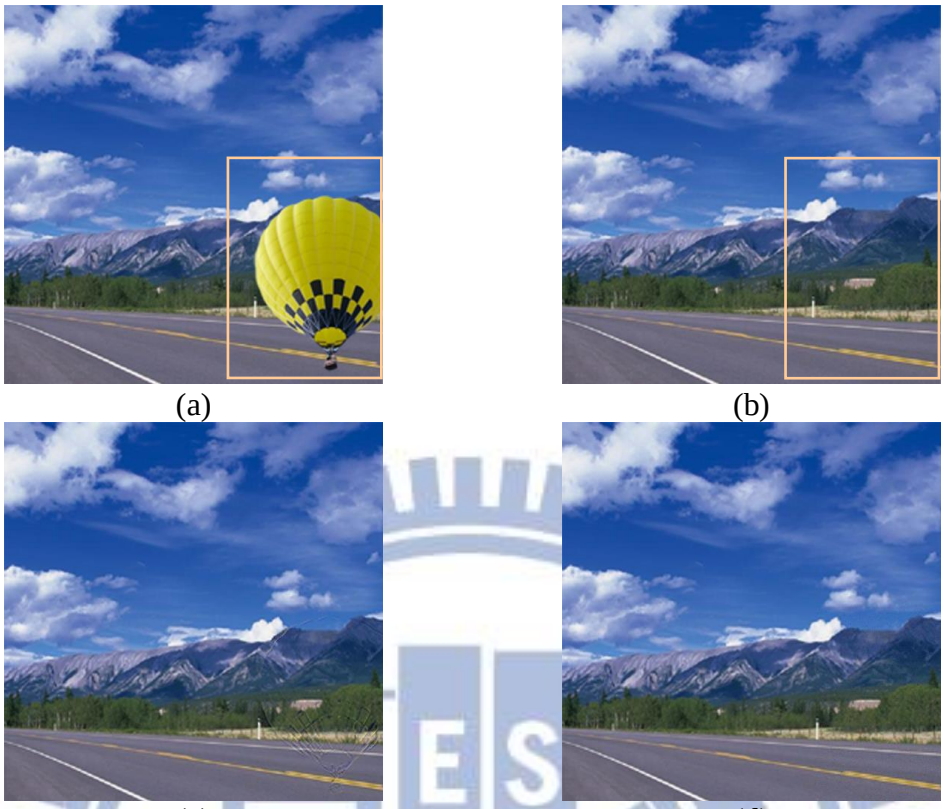


Figure 3.6 Example of quality improvement results by Lin and Tsai [25]. (a) Privacy-sensitive image part. (b) Pre-selected background image part. (c) Camouflage image generated from (a) and (b) by using prediction-based mapping. (d) Camouflage image generated from (a) and (b) by quality improvement.

Instead, Lin and Tsai [25] proposed to take the sum of the values of the three adjacent pixels of each pixel P (excluding itself) in the prediction-residue image I_r as the side information M of P, as illustrated in Figure. 3.7, and then decided whether to reduce a half of the residue value of P or not, as illustrated in Figure. 3.8 — if M is larger than a pre-selected threshold, then the value reduction operation is conducted. This scheme of using the side information is reasonable because the values of the pixels adjacent to the edge pixels in the prediction-residue image will be large as well in general.

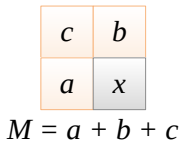


Figure 3.7 Computing the side information M of the current pixel value x.

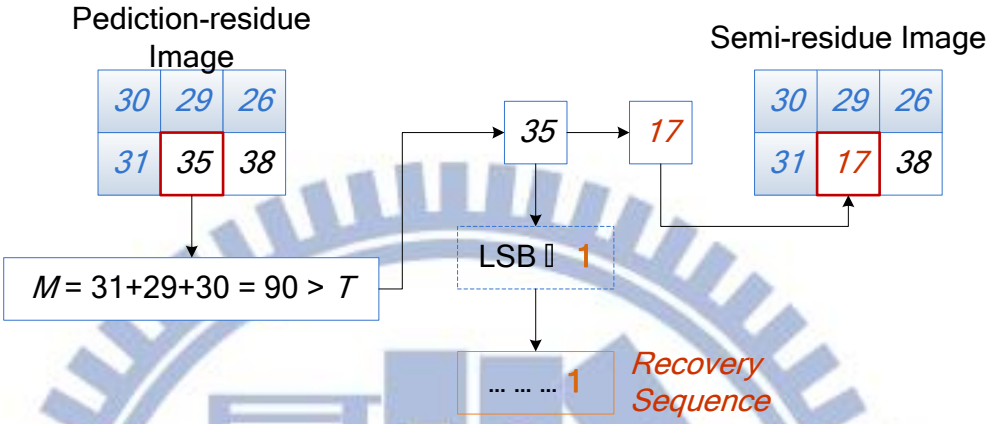


Figure 3.8 An example of reducing prediction residue by Lin and Tsai [25].

Reduction of the residue value to a half is accomplished by removing the LSB of pixel P, called the removed bit. This removed bit then is recorded into a recovery sequence L_R. These steps are applied to each of the three color channel. When all pixels in the prediction-residue image I_r are processed, a new prediction-residue image I_r' with smaller pixel values is obtained, which we call a semi-residue image. Furthermore, we embed the resulting recovery sequence L_R into the resulting camouflage image by the RCM scheme [32], which is a lossless LSB-modification scheme, for the purpose of later recovery of the original privacy-sensitive image part I_s.

Stage 2 – randomizing semi-residue image content

On the other hand, as seen in Figure 3.6(c), the larger values of the semi-residue image will appear roughly as edge contours in the final camouflage image. It is desired to reduce such a phenomenon in order to mitigate attraction of human visual attention for the security purpose. It seems that for this aim we may eliminate such

edge contours from the semi-residue image before the image is mapped to compose the camouflage image. However, Lin and Tsai [25] did not do so because the data of the semi-residue image should be kept intact for later privacy-sensitive image recovery. For this reason, instead they randomized the positions of the pixels in the semi-residue image with a secret key before the pixel values are mapped to compose the camouflage image. In such a way, the resulting camouflage image includes much less edge contours and looks much like the background image frame, as revealed in the experimental results. An example is shown in Figure 3.6(d). Furthermore, any user without the same secret key cannot recover the original semi-residue image, and so is not able to retrieve the privacy-sensitive image part.

3.3.2 Proposed process of privacy-sensitive region concealment

Now, it is time to describe the entire process of privacy-sensitive video content concealment. An illustration is shown in Figure 3.9. First, an authorized user needs to specify a protected region *R* in an input surveillance video, and the video content in *R* is defined as the privacy-sensitive image part. Then, Lin and Tsai [25] applied the concealment process with a pre-selected background image part (in the first frame of the video) to the privacy-sensitive image part in each image frame in the video to produce a protected image.

When all privacy-sensitive images have been completed, Lin and Tsai [25] embedded the start and end positions of the protected region *R* into the pre-selected background image which was the first image in the surveillance video. After these steps, a protected privacy video have been obtained. Both the positions of the protected region and the recovery sequence are used to recover the privacy-sensitive

image later. The reason of using them will become clear in the recovery algorithm which is describes later.

In order to display the 3D image information, we have proposed a method of merging color and depth images acquired by the KINECT device, as described by Algorithm 3.1. Because we calibrate and map the color and depth image first, synchronization of the removals of corresponding areas in the color and depth images is guaranteed.

On the other hand, we have proposed a method to remove indoor specific space privacy and user selected privacy-sensitive region, and use the environmental data to replace the privacy-sensitive region to achieve the function of concealment. In this method, the original privacy-sensitive region will be hidden, and unauthorized users can only see the 3D surveillance video without the privacy-sensitive information. But, an authorized user can recover the original data by the secret key.

The method proposed by Lin and Tsai [25] were used to remove privacy information in 2D surveillance videos. The method develop in this study extends their method to 3D surveillance videos, by developing related techniques of detecting, inpainting, and recovering the specific privacy-sensitive region.

The following is the algorithm implementing our idea of privacy concealment. A flow chart is given in Figure 3.9, as mentioned. After we used the depth and color images to achieve privacy concealment, we will also recover the depth and color information, the algorithm implementing our idea of privacy recovery as mentioned in section 3.3.3.

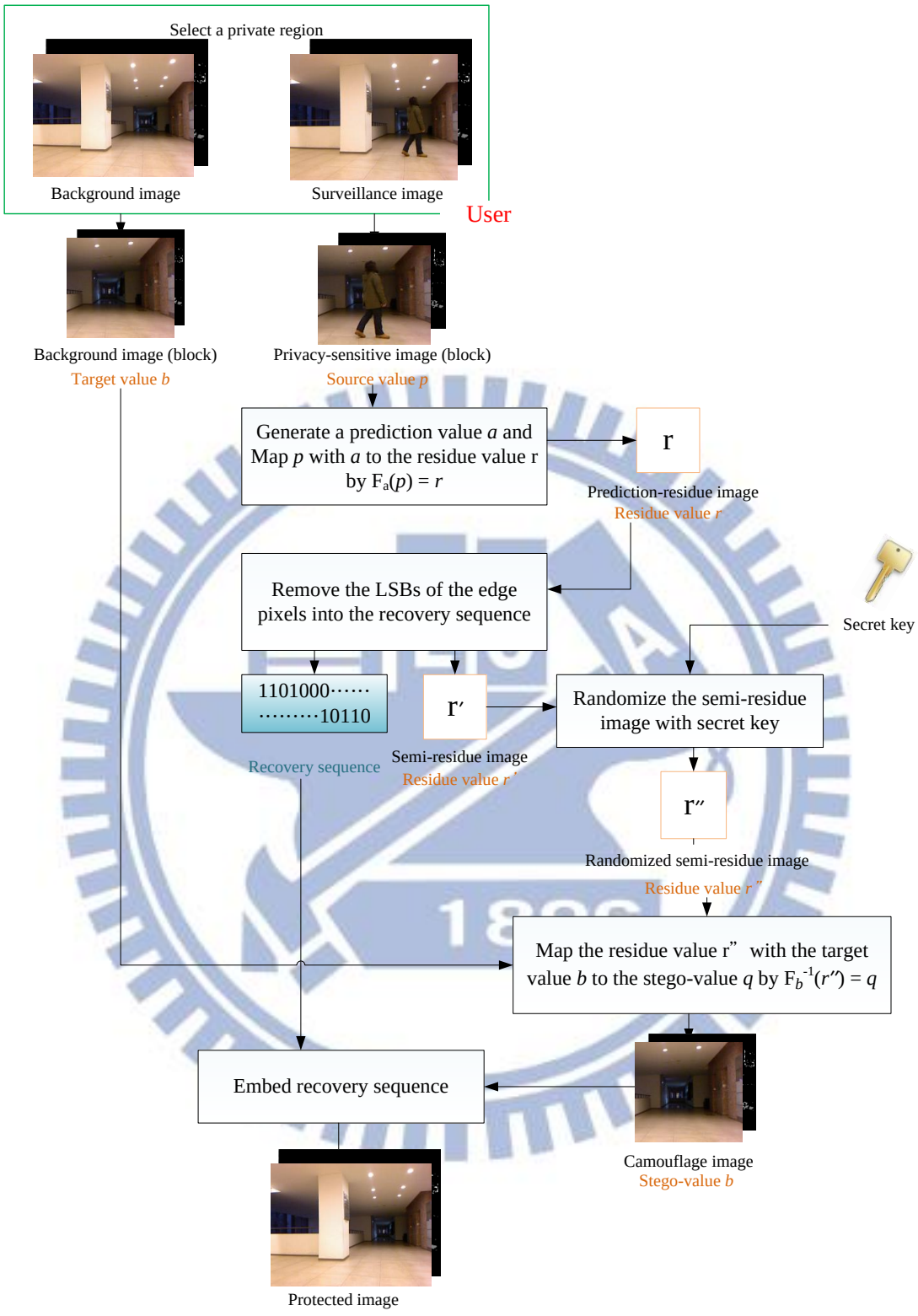


Figure 3.9 Flowchart of the proposed privacy-sensitive region concealment process.

Algorithm 3.6: process for the privacy-sensitive image concealment.

Input: a privacy-sensitive color image frame V_c, a privacy-sensitive depth image frame V_d, a background color image frame B_c, a background depth image frame B_d, a secret key K, a threshold value T, and a region R specified by an authorized user.

Output: a protected color image frame V_c'' and a protected depth image frame V_d'' with the privacy information in R being concealed.

Steps.

Stage 1 --- producing a prediction-residue image from the privacy-sensitive image.

Step 1. Initialize a prediction-residue image E with the same size of R with the pixel value all set to be (0, 0, 0).

Step 2. For each pixel P of V_c and V_d in the specified region R, perform the following steps for each color channel and for the depth channel in a raster-scan order to derive the prediction residue value of pixel P and record the result in E.

2.1 Get the prediction value a of the pixel according to Algorithm 3.5.

2.2 Set p to be the value of P.

2.3 Map p with a to a new value r according to $F_a(p)$ in Algorithm 3.4.

2.4 Set the value of the pixel in E corresponding to P in V_c and V_d to be r.

Stage 2 --- Generating the randomized semi-residue image before mapping.

Step 3. Initialize a recovery sequence L_R.

Step 4. Initialize a semi-residue image E' with the same size of E with the pixel value all set to be (0, 0, 0) initially.

Step 5. For each pixel P' in E, perform the following steps for each color channel and the depth channel in a raster-scan order to decide whether the LSB of

the value of P' need be removed or not.

5.1 Set the value of the corresponding pixel in E' to be the value C_p of P'.

5.2 Compute the sum h of the three adjacent pixel values of P' in E in a way as illustrated in Figure 3.6.

5.3 If h is larger than the threshold T, perform the following steps.

5.3.1 Remove the LSB of C_p into L_R.

5.3.2 Compute the remaining value d of P' by $d = \left\lfloor \frac{C_p}{2} \right\rfloor$.

5.3.3 Reset the value of the pixel in E' corresponding to P' in E to be d.

Step 6. Rearrange the position of the pixel value in E' randomly by the secret key K, resulting in a randomized semi-residue image E''.

Stage 3 --- Producing the protected surveillance image.

Step 7. For each pixel P' of V in R, perform the following steps for each color channel in a raster-scan order to produce a camouflage image V_c' and V_d' by mapping the value of the pixel in E'' and that of the pixel in B_c and B_d, both corresponding to P' in V_c and V_d, to get a stego-value q.

7.1 Denote the value of the corresponding pixel in E'' as r and that in B_c and B_d as b.

7.2 Map r with b to a new value q according to $F_b^{-1}(r)$ in Algorithm 3.5.

7.3 Set the value of the pixel to be q.

Step 8. Perform the following steps to embed the length l_L of the recovery sequence L_R into the resulting camouflage image V_c' and V_d'.

8.1 Transform the length l_L of L_R into a bit string S_R (with length l_S).

8.2 Compute the maximum number N_L of bits for embedding S_R by the following equation:

$$N_L = \left\lceil \log_2 \left(W_V \times H_V \right) \right\rceil \tag{3.3}$$

where W_V and H_V are the width and height of V', respectively.

8.3 Prefix 0's to the beginning of S_R to compose a string of length N_L (with $N_L - l_S$ leading 0's).

8.4 Embed the first unembedded bit b_s of S_R into a pixel in V_c'' and V_d'' (with all bits in S_R regarded as unembedded initially).

8.5 Repeat Step 8.3 N_L times.

Step 9. Perform the following steps to embed L_R into V_c' and V_d'.

9.1 Embed the first unembedded bit b_l of L_R into a pixel in V_c' and V_d' (with all bits in L_R regarded as unembedded initially).

9.2 Repeat Step 9.1 until reaching the end of L_R.

Step 10. Output the resulting V_c' and V_d' as the desired protected surveillance images V_c'' and V_d''.

3.3.3 Proposed process for privacy-sensitive region recovery

In this section, we describe how to retrieve the privacy-sensitive image frame from the protected image using the method of Lin and Tsai [25]. The recovery process is illustrated by Figure 3.10. Before recovering each privacy-sensitive image frame, we need to extract the start and end positions of the protected region from the background image part and recover the original background image frame by the lossless RCM scheme [32] in order to retrieve the randomized semi-residue image from the original camouflage image and the original background image frame.

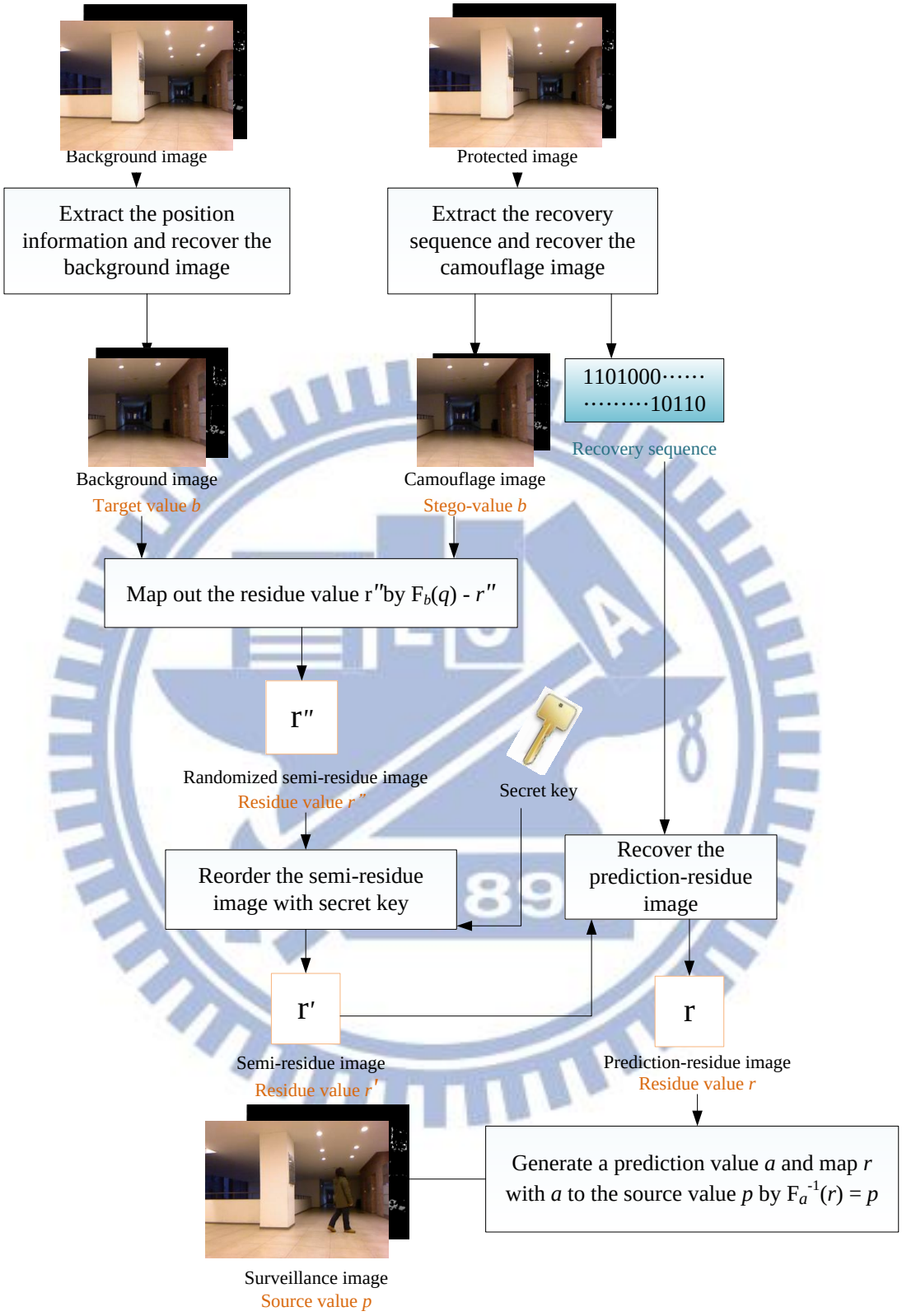


Figure 3.10 Flowchart of the proposed privacy-sensitive region recovery process.

Algorithm 3.7: Process for the privacy-sensitive image recovery.

Input: a protected color image frame Q_c, a protected depth image frame Q_d, a pre-selected background color image frame B_c, a pre-selected background depth image frame B_d, a secret key K, a threshold value T, and a specified region R.

Output: the original privacy-sensitive image frame Q' recovered from Q_c and Q_d.

Steps:

Stage 1 --- Retrieve the randomized semi-residue image from the protected image.

Step 1. Get the number N_L of bits for extracting the recovery sequence L_R by Equation 3.3, where W_V and H_V are set to be the width and height of Q_c or Q_d, respectively.

Step 2. Perform the following steps to extract the length l_L of L_R from Q_c and Q_d.

2.1 Select an *unprocessed* pixel pair T_i from Q_c and Q_d in a raster-scan order (with all pixel pairs in Q_c and Q_d regarded as unprocessed initially).

2.2 Extract an LSB from T_i by an inverse version of the lossless RCM scheme [32], append it to a string B_i (initially empty), and in the meantime, decrease N_L by 1.

2.3 If $N_L \neq 0$, then go to Step 2.2; otherwise, continue.

2.4 Transform B_i obtained in Steps 2.2 into a decimal value and take it as the value of l_L.

Step 3. Perform the following steps to extract the recovery sequence L_R from Q_c and Q_d.

3.1 Select an *unprocessed* pixel pair T_i from Q_c and Q_d in a raster-scan order.

3.2 Extract an LSB from T_i by an inverse version of the lossless RCM scheme [32], append it to the recovery sequence L_R (initially empty), and in the meantime, decrease l_L by 1.

3.3 If $l_L \neq 0$, then go to Step 3.2; otherwise, continue.

Step 4. For each pixel P of Q_c and Q_d in region R, perform the following steps to retrieve the randomized semi-residue image S.

4.1 Set q to be the value of P.

4.2 Set b to be the value of the pixel in B corresponding to P in Q_c and Q_d.

4.3 Set the value r of the pixel in S corresponding to P in Q_c and Q_d by $F_b(q) = r$ according to Algorithm 3.4.

Stage 2 --- Recover the original prediction-residue image S'' from S

Step 5. Reorder the positions of the pixel values in S by the secret key K, resulting in an image called the original semi-residue image S'.

Step 6. For each pixel P' of S', perform the following steps to recover the original value of the corresponding pixel in S''.

6.1 Compute the sum h of the three adjacent pixel values in S' of P' in a way as illustrated in Figure 3.6.

6.2 If h is larger than T, perform the following steps.

6.2.1 Extract the first bit b_s in L_R.

6.2.2 Double the value of P'.

6.2.3 Add b_s to the value of P'.

Stage 3 --- Recover the privacy-sensitive image Q' from S''.

Step 7. Set the value of each pixel in the desired privacy-sensitive image Q_c' and Q_d', which is outside region R, to be equal to that of the corresponding pixel in Q_c and Q_d, respectively.

Step 8. For each pixel P'' of Q_c' and Q_d' within R, perform the following steps to recover the original value.

8.1 Obtain the same prediction value a as that derived in Step 2.2.1 of Algorithm 3.7 by applying Algorithm 3.6.

8.2 Set r' to be the value of the pixel in S'' corresponding to P'' in Q_c' and Q_d'.

8.3 Restore p from r' by setting $p = F_a^{-1}(r')$.

8.4 Set the value of P'' to be p.

3.4 Experimental Results

In this chapter, we have proposed a method for protecting privacy-sensitive region in a 3D surveillance video using the prediction-based mapping [25] and adding the depth image acquired by KINECT device, which allows us to merge the color and depth images. Based on the technique, we can convert a privacy-sensitive image into a camouflage image which looks similar to the background image, and synchronize removals of corresponding areas in color and depth images.

With this proposed privacy protection system, we can conceal the privacy parts of 3D surveillance video contents to avoid legal disputes and to protect the personal privacy of non-suspicious people.

Some experimental results of applying the proposed method for protecting privacy-sensitive regions in a surveillance video are shown in Figures 3.11 through 3.19. First, we protected privacy-sensitive regions in the color image sequences, and later recovered the original protected color images. The results are shown in Figures 3.11 through 3.13. Next, we protected the corresponding privacy-sensitive regions in the depth images, and recovered the original protected depth images. The result are

shown in Figures 3.14 through 3.16. At last, we used algorithm 3.1 to merge the color and depth images to display the 3D surveillance video. The result are shown in Figures 3.17 through 3.19.

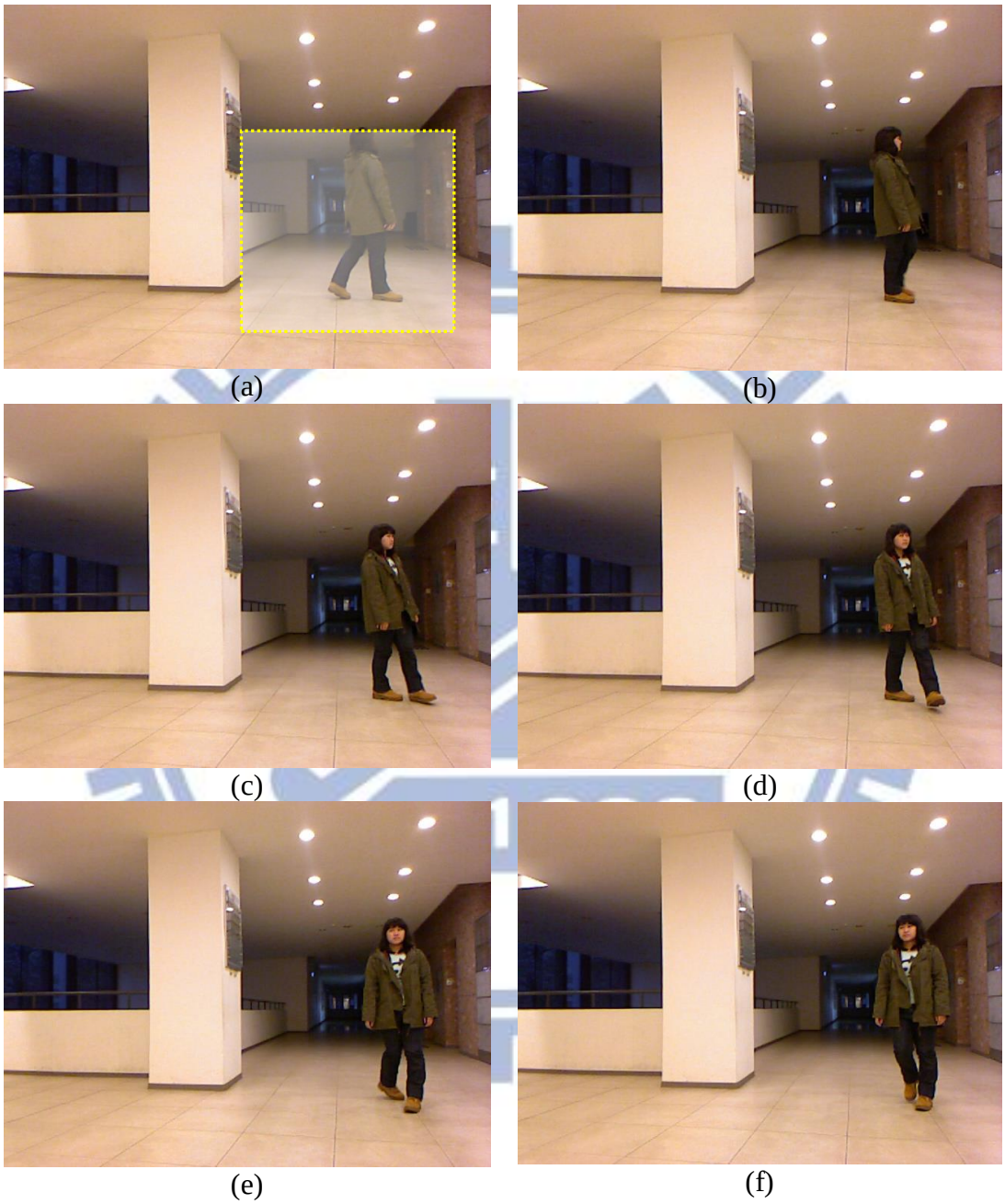
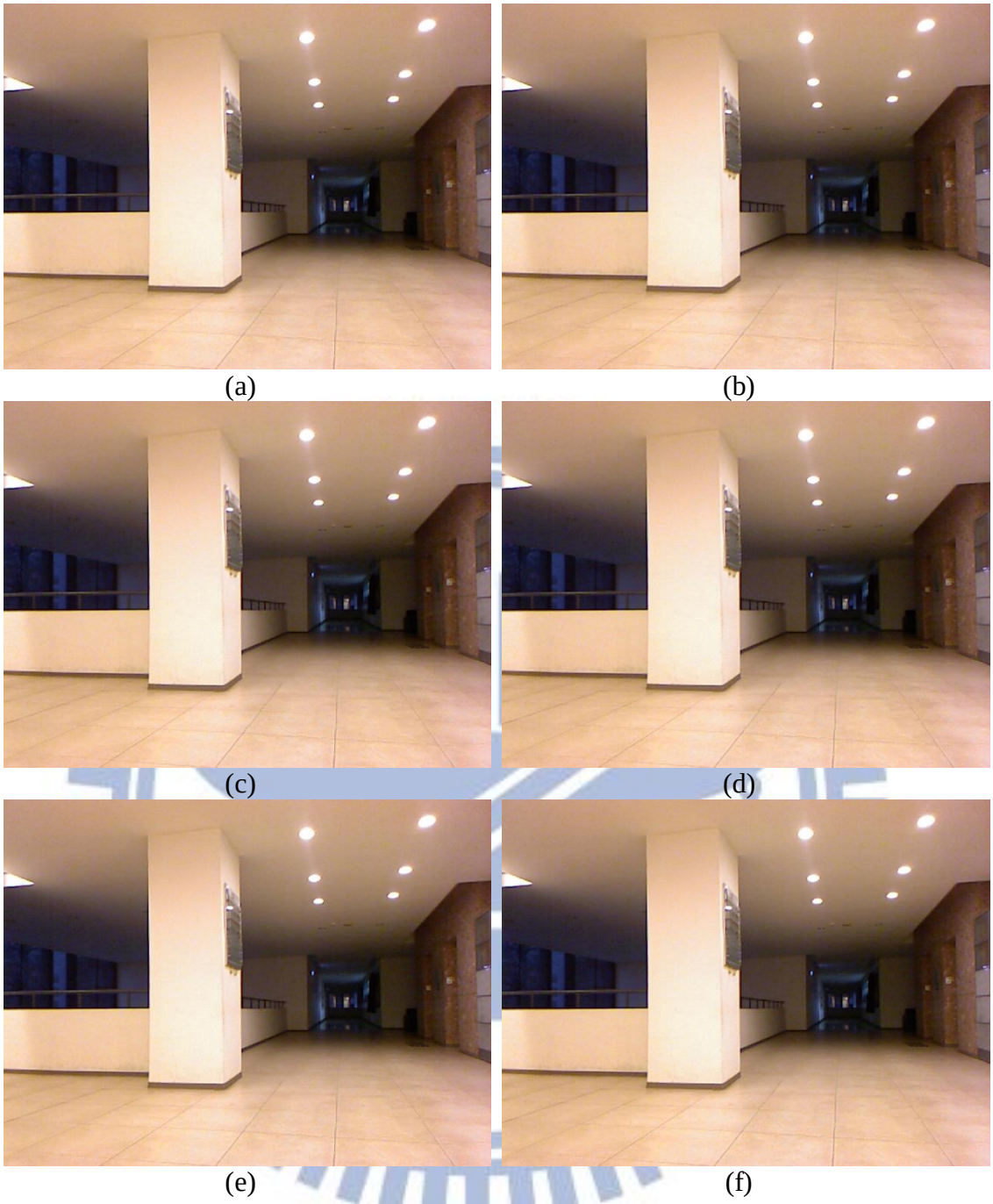


Figure 3.11 Six representative frames of a color surveillance video. (a) The 18th frame with the protected region enclosed by a rectangle. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.



(a) (b) (c) (d) (e) (f)

Figure 3.12 Six representative frames of a privacy-protected color video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

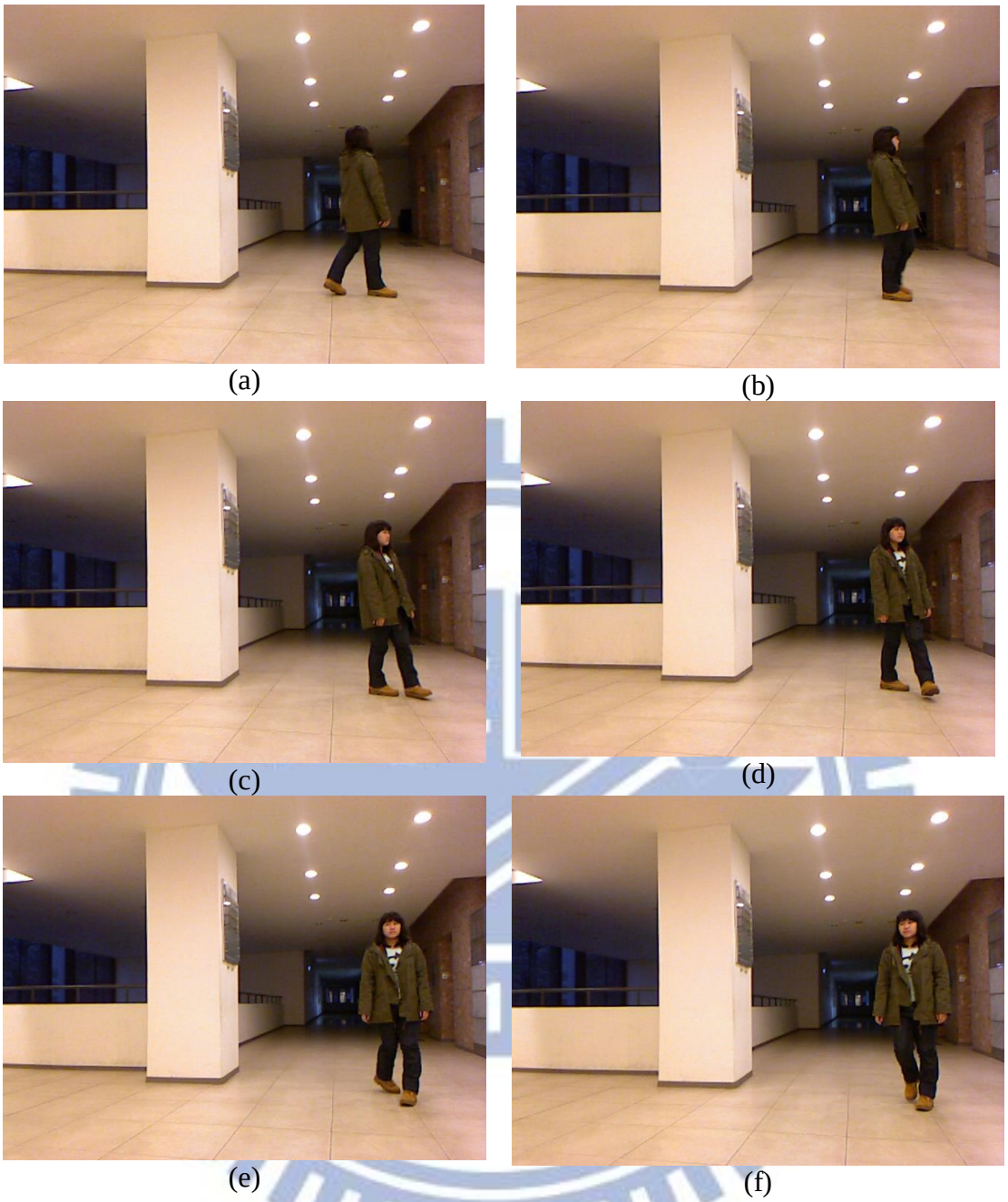


Figure 3.13 Six representative frames of the recovered video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

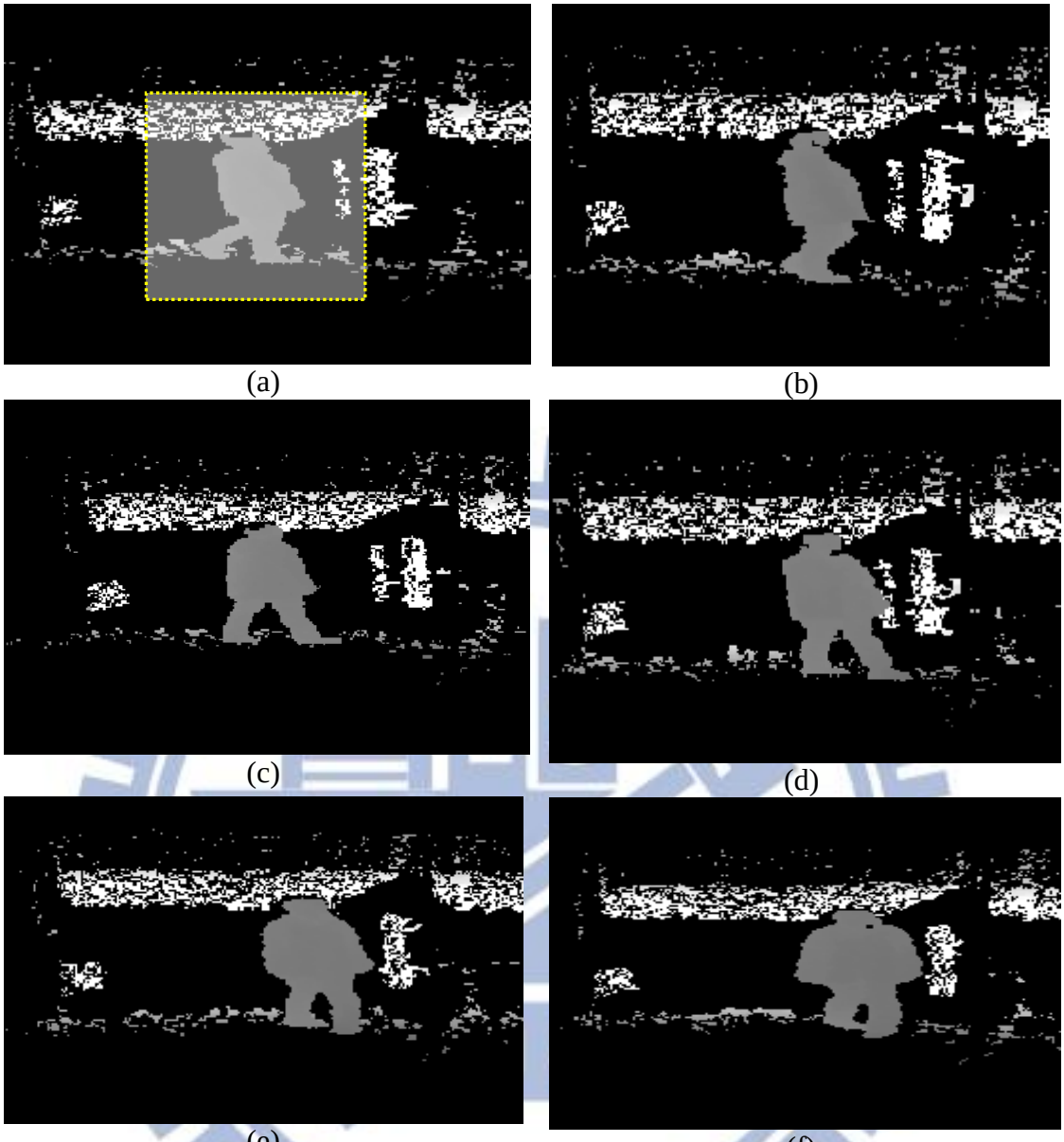


Figure 3.14 Six representative frames of the depth surveillance video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

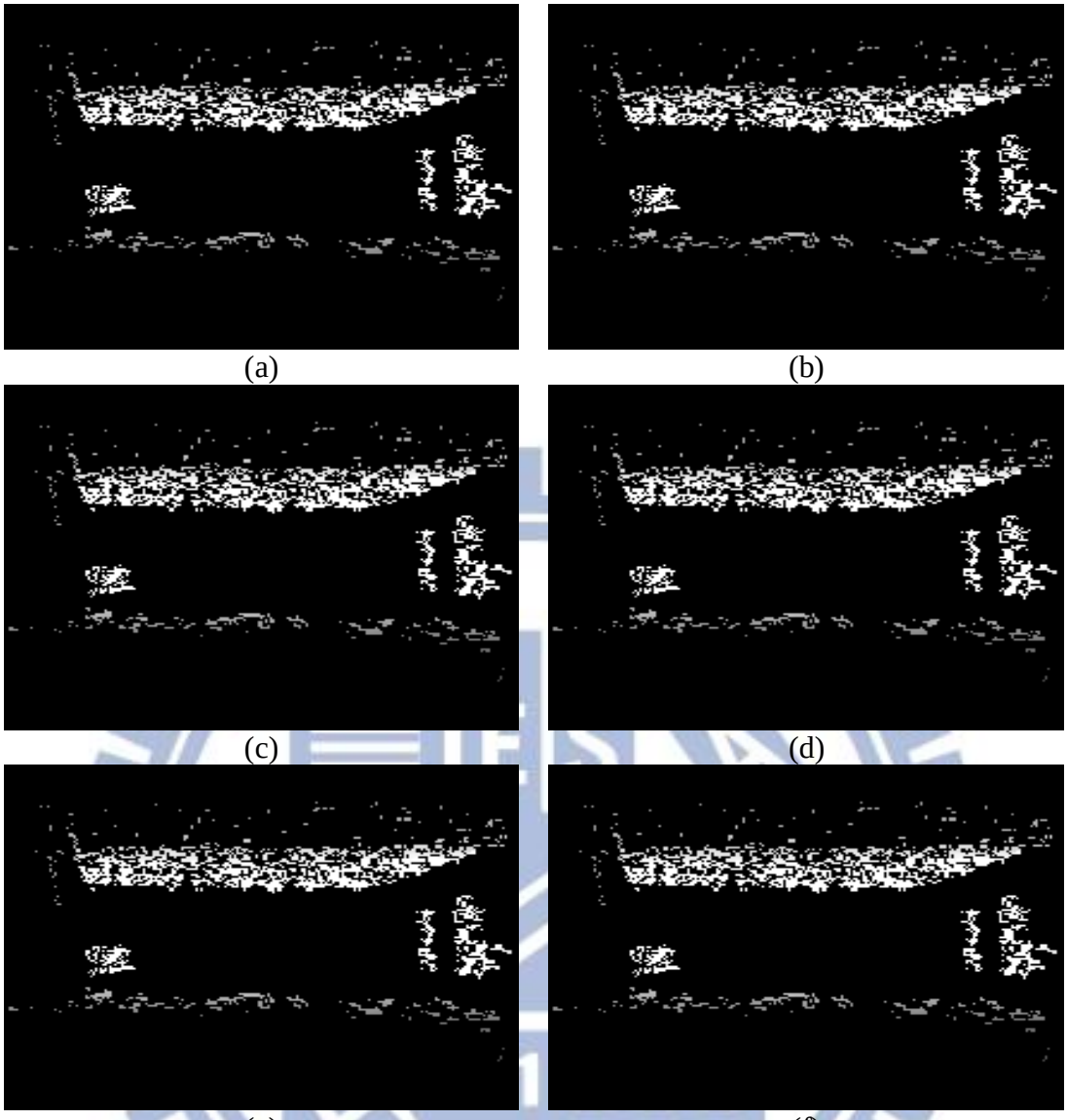


Figure 3.15 Six representative frames of the privacy-protected depth video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

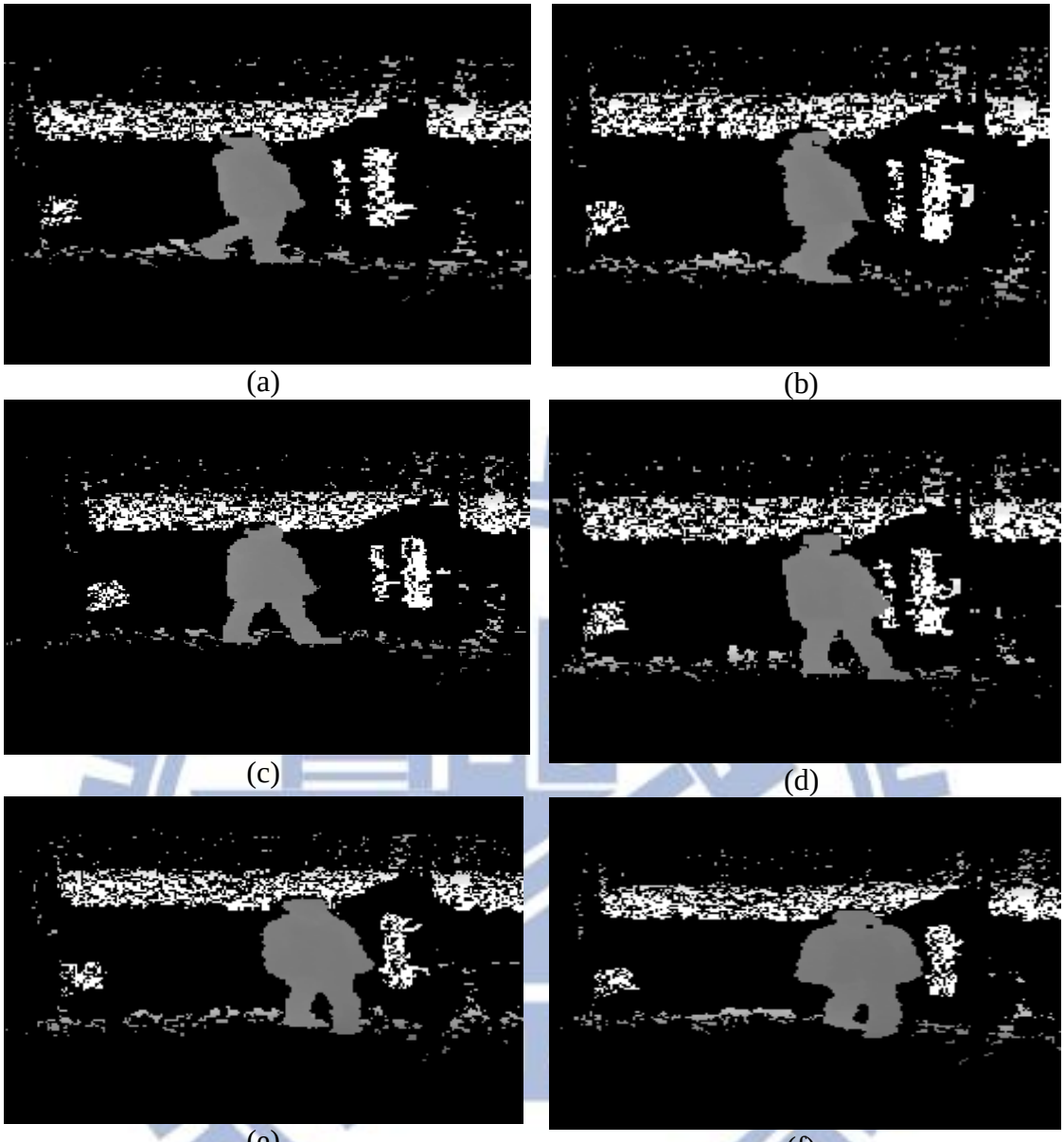


Figure 3.16 Six representative frames of the depth recovered video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

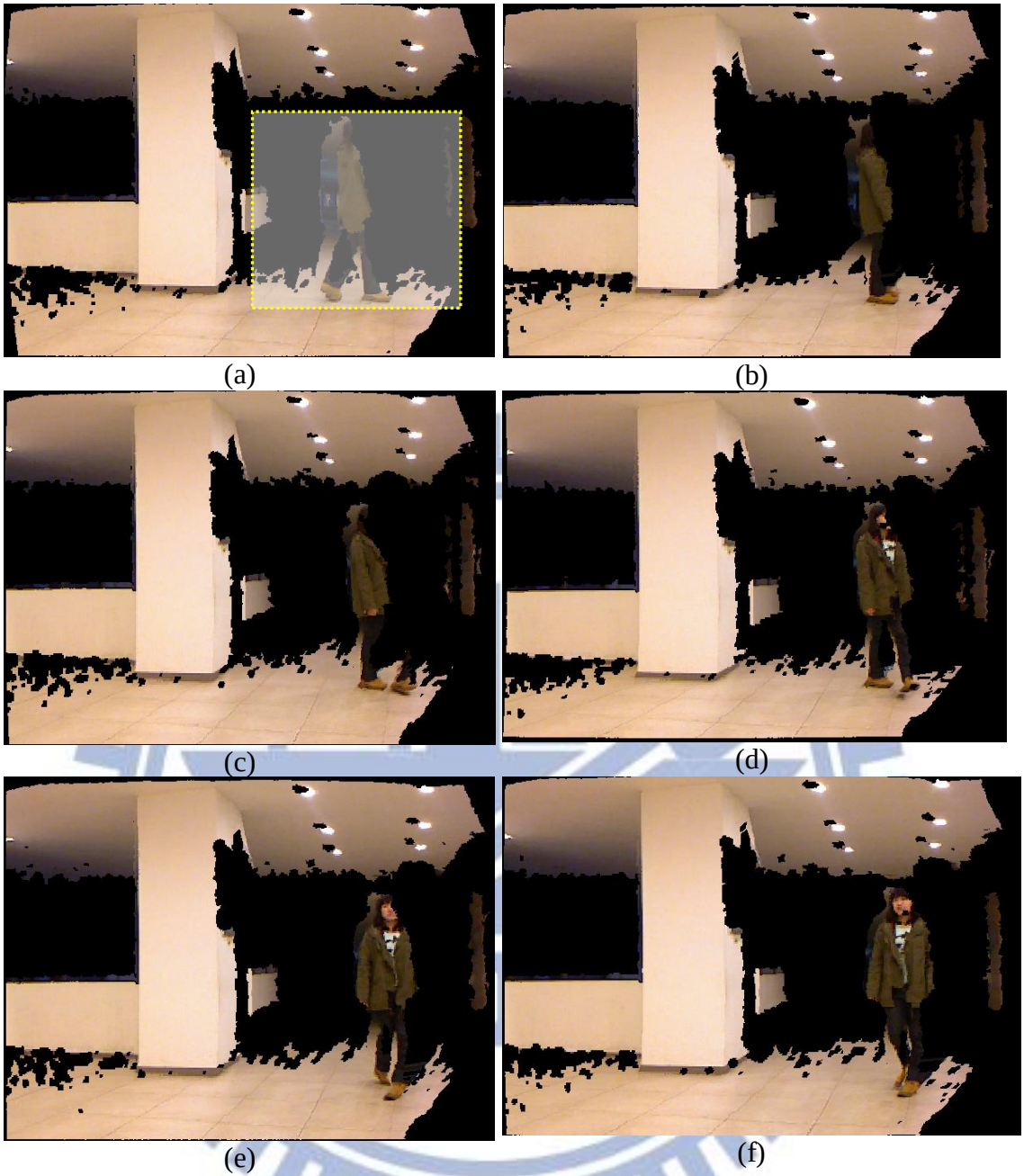


Figure 3.17 Six representative frames of a 3D surveillance video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

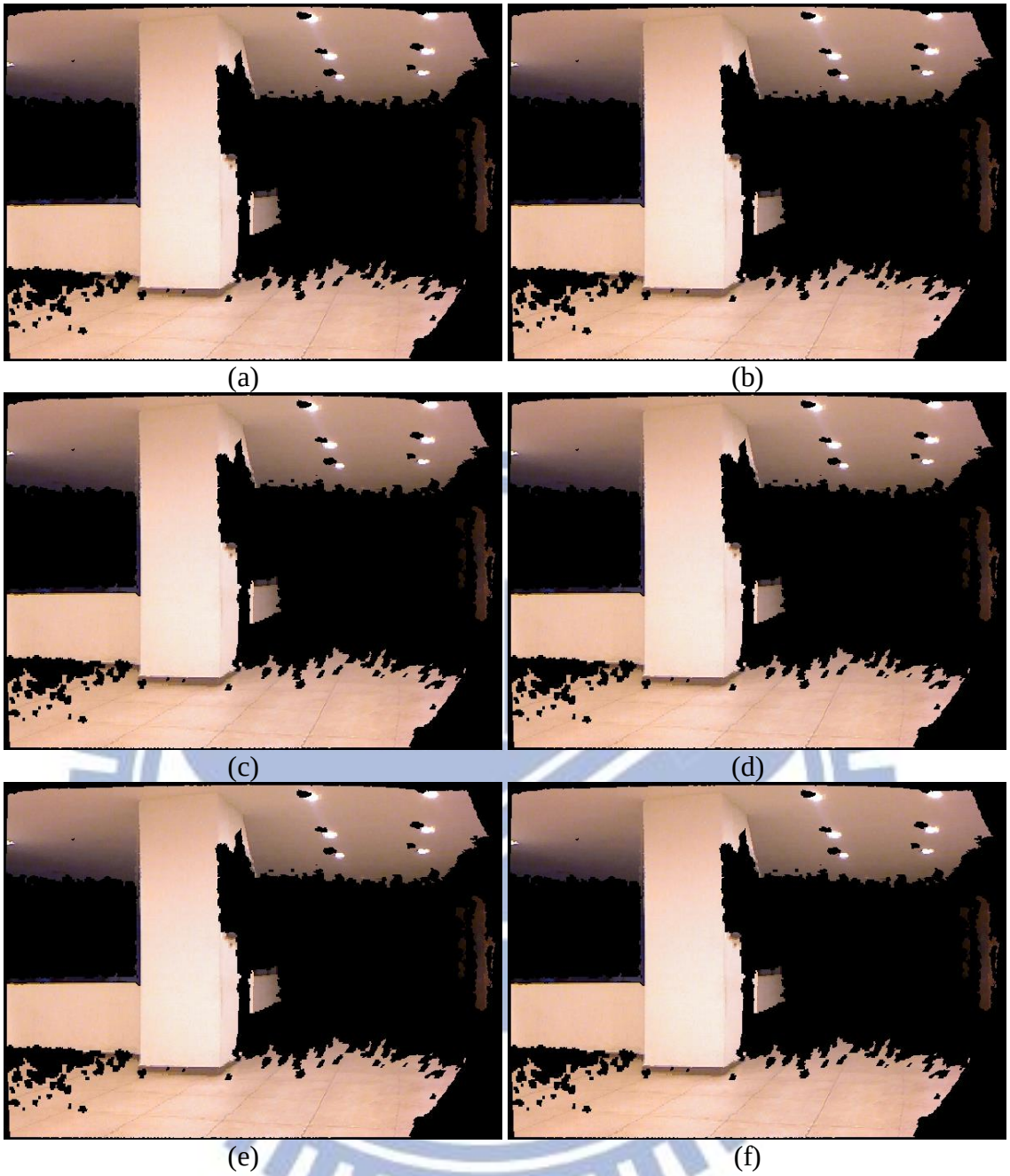


Figure 3.18 Six representative frames of a 3D privacy-protected video. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

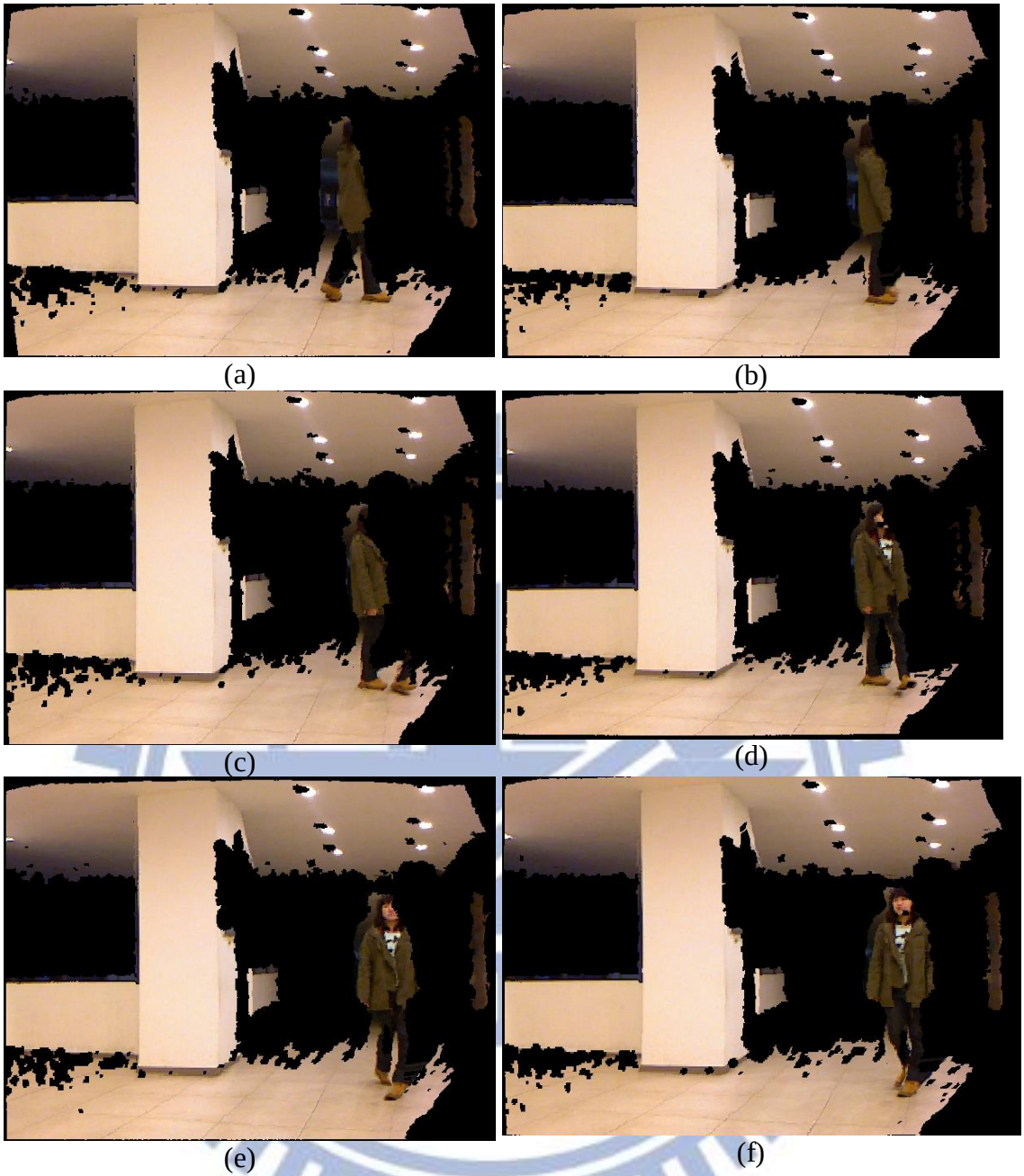


Figure 3.19 Six representative frames of a 3D recovered video combining the previously-shown color and depth images. (a) The 18th frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

Chapter 4
Protection of Privacy-sensitive Motion Activities in Surveillance Videos Acquired by KINECT Devices

4.1 Introduction

The proposed method for region-based privacy protection, which can be utilized for protecting selected privacy-sensitive regions in KINECT images, is described in the previous chapter. Specifically, we use a portion of a pre-selected background image to cover a selected privacy-sensitive region in an image frame of a surveillance video by a reversible prediction-based mapping scheme [25]. It is desirable that the proposed technique can be applied to conceal privacy-sensitive motion activities and that motion activities can be detected by the use of speed up robust features (SURFs). The proposed method for these purposes will be described in this chapter. In Sections 4.1.1 and 4.1.2, the related problem definitions and a review of the ideas of the proposed method are given. The detailed algorithms of motion-activity concealment and recovery are presented in Section 4.2. Finally, some experimental results showing the feasibility of the method are given in Section 4.3.

4.1.1 Problem definition

The previous chapter is about the protection of a *location-fixed* privacy-sensitive region in each image frame of a video. A second application is the protection of privacy-sensitive *motion activities* in a video. In such an application, the location of the privacy-sensitive region is not fixed, but changes in accordance with the location

of the motion activity appearing in each frame.

In the problem of protecting privacy-sensitive motion activities in videos dealt with in this study, we first select a region for detecting motion events in each image frame of the surveillance video automatically. Then, we try to apply the previously-proposed process of the privacy-sensitive region concealment [25] (described in Chapter 3) to the selected region in each video frame. We can get the difference between the pre-selected background image (the first frame of the video) and currently-processed surveillance image frame by using SURFs.

In addition, the proposed scheme to solve this problem must allow the concealed privacy-sensitive image part to be retrieved losslessly. Also, the depth image and the color image taken by the KINECT device at an identical instant of time should be protected together to keep their relation in time, as mentioned before.

4.1.2 Review of Ideas of a Previous Study

In this study, we still adopt the reversible prediction-based mapping from the previous study of Liu and Tsai [1] and Lin and Tsai [25]. The method for privacy protection in surveillance videos is based on the use of the reversible prediction-based mapping proposed by Liu and Tsai [1], which is a deterministic one-to-one compound mapping of values. Besides, Lin and Tsai [25] proposed a principle of mapping which is used for protection of pre-selected privacy-sensitive regions. And we will use the algorithm 3.1, which is mentioned in chapter 3, to merge the color and depth images to display 3D information.

4.2 Proposed Method for Protecting Privacy-sensitive Motion Activities in Surveillance Videos

In this section, we describe in detail the proposed method for protecting privacy-sensitive motion activities in surveillance videos. In Section 4.2.1, the proposed method for detection of motion activities by the use of speeded up robust features (SURFs) is described. In Section 4.2.2, the proposed method of the previously-described privacy concealment process for use in protecting privacy-sensitive motion activities is described. In Section 4.2.3, the process of motion-activity recovery is presented.

4.2.1 Detection of Motion Activities by Use of Speeded Up Robust Features (SURFs)

In the proposed method for protecting privacy-sensitive motion activities, at the beginning we search motion activities in videos, and then decide a corresponding protected region *R* for detecting motion event automatically. Also, the image content in *R* is defined as a privacy-sensitive image part and will be disguised as a pre-selected background image part *B* which corresponds to the privacy-sensitive image part in position in the first image frame of the video. Then, we apply the previously-proposed concealment process (described in Chapter 3) with the background image part to the privacy-sensitive image part to produce a camouflage image looking close to the background image part *B*.

Before we start, we briefly introduce the principles of SURFs (speeded up robust features) and how to detect motion activities by use of SURFs. SURFs are *robust*

local features, first presented by Herbert Bay et al. [33] in 2006, that can be used in computer vision tasks like object recognition or 3D reconstruction. It is partly inspired by the SIFT descriptor. The SURFs have been proven to achieve high repeatability and distinctiveness. The method using SURFs uses a Hessian matrix-based measure for the detection of interest points and a distribution of Haar-wavelet responses within the interest point neighborhood as a descriptor. An image is analyzed at several scales, so interest points can be extracted from both global and local image details. Therefore, the SURF extraction and matching scheme are one of the best interest point detectors and descriptors currently available.

(A) Extraction of feature points

First, based on the good performance in computation time and accuracy of the Hessian matrix, the matrix is to determine the location and scale of the SURF descriptor. Given a point $x = (x, y)$ in an image I, the Hessian matrix $H(x,\sigma)$ in x at scale σ is defined as follows:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{4.1}$$

where $L_{xx}(x,\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I in point x, and $L_{xy}(x,\sigma)$ and $L_{yy}(x,\sigma)$ are similarly interpreted.

Next, the SURFs approximate second order derivatives of the Gaussian with box filters. Image convolutions with these box filters can be computed rapidly by using integral images. The determinant of the Hessian matrix is written as:

$$Det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{4.2}$$

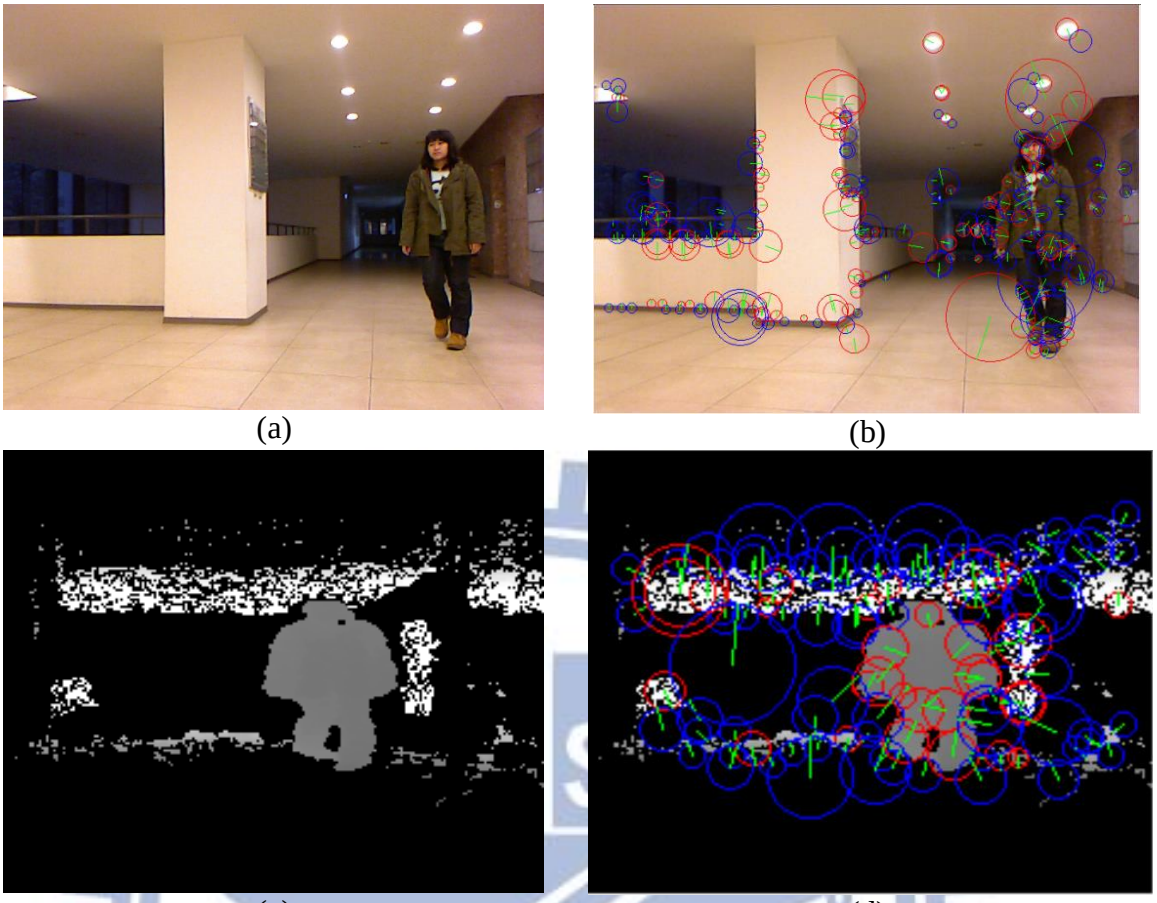
In order to localize interest points in the image and over scales, a non-maximum

suppression in a 3x3x3 neighborhood is applied. Finally, the found maxima of the determinant of the Hessian matrix are then interpolated in the scale and image space.

(B) Descriptor of feature extraction

The SURF descriptor is extracted from an image in two steps: the first step is assigning an orientation based on the information of a circular region around the detected interest points. The orientation is computed using Haar-wavelet responses in both *x* and *y* direction. Once the Haar-wavelet responses are calculated and weighted with a Gaussian ($\sigma = 2.5s$) centered at the interest points. In a next step the dominant orientation is estimated by summing the horizontal and vertical wavelet responses within a rotating wedge which covers an angle $\pi/3$ in the wavelet response space. The resulting maximum is then chosen to describe the orientation of the interest point descriptor.

In the method proposed in this study, at first we segment respective privacy-sensitive image from currently-processed surveillance images automatically. Next, we extract feature points from these images using the SURF extraction equations (4.1) and (4.2), and then find the descriptor of the feature points in the currently-processed surveillance images. At last, the feature points in the currently-processed surveillance image shown in Figure 4.1, where the size of the circle specifies the scale and the line in the circle is the orientation of the feature point.



(a) (b)

(c) (d)

Figure 4.1 Feature extractions from surveillance image. (a) The color-image of the 23th frame (b) Feature points of the color-image of the 23th frame (c) The depth-image of the 23th frame (d) Feature points of the depth image of the 23th frame.

After extracting feature points from both the privacy-sensitive image and the currently-processed surveillance images, we compare the motion object frame by frame. In this way, we can find the feature points of the object specifically, and won't be affected by other factors. Therefore, when we obtain the matched feature points, we try to find a bounding box to include them. As shown in Figure 4.2, the red box is the bounding box of the matched feature points and the motion object. Finally, we obtain the region of each motion object, and match the privacy-sensitive image and currently-processed surveillance images together.

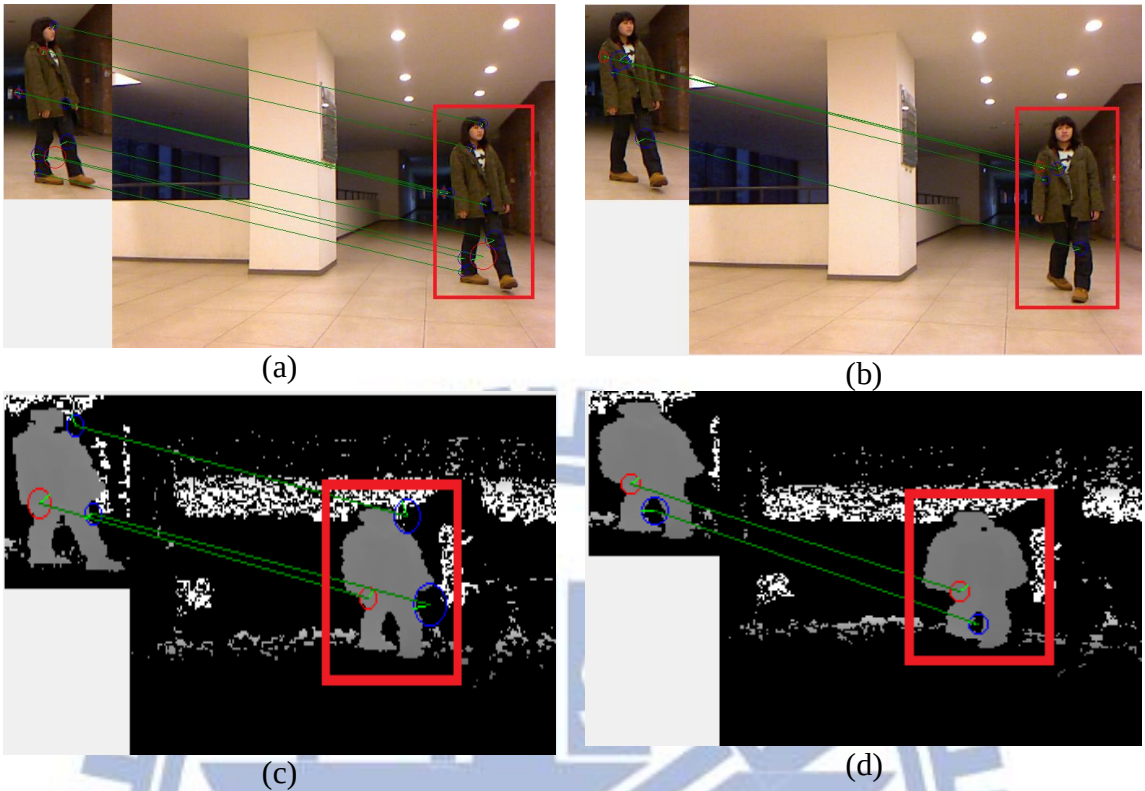


Figure 4.2 A result of the proposed method for detecting moving objects by use of SURFs. (a) The color-image of the20th frame compare with 21th frame. (b) The color-image of the 21th frame compare with 22th frame. (c) The depth-image of the 20th frame compare with 21th frame. (d) The depth -image of the 21th frame compare with 22th frame.

The following algorithm describes the process to match feature points between privacy-sensitive images and currently-processed surveillance images.

Algorithm 4.1: Detecting motion activities, and matching the privacy-sensitive image and currently-processed surveillance images by the use of SURFs.

Input: A surveillance color image sequence $\{S_{c1}, S_{c2}, \ldots, S_{cn}\}$ and a surveillance depth image sequence $\{S_{d1}, S_{d2}, \ldots, S_{dn}\}$; and an initial surveillance color image S_{c0} and an initial surveillance depth image S_{d0}, both being rectangular in

shape with width w_0 and height h_0 and including an identical object (a human).

Output: A privacy-sensitive color image sequence P_{c1}, P_{c2}, …, P_{cn} and a privacy-sensitive depth image sequence P_{d1}, P_{d2}, …, P_{dn}.

Steps:

Step 1. For $i = 0, 1, 2, \ldots, n-1$, conduct the following the follows steps.

1.1 Extract a feature point set F_i from P_{ci} by the SURF extraction algorithm mentioned previously.

1.2 Extract a feature point set W_{i+1} from $S_{c(i+1)}$ by the SURF extraction algorithm mentioned previously, respectively.

1.3 Match F_i with W_{i+1} to obtain as a *match set* F_i' those feature points in W_{i+1} which have corresponding feature points in F_i.

1.4 Find the feature point f_i' in the match set F_i', which has the minimum Euclidean distance D_{i+1} to the origin o_i of P_{ci} and whose corresponding feature point in W_{i+1} is w_{i+1}.

1.5 Use the distance D_{i+1} between f_i' and o_i to find the corresponding points c_{i+1} and d_{i+1} of w_{i+1} from $S_{c(i+1)}$ and $S_{d(i+1)}$, respectively.

1.6 Find the regions of $P_{c(i+1)}$ and $P_{d(i+1)}$ with origins c_{i+1} and d_{i+1}, respectively, both with width w_i and height h_i, in $S_{c(i+1)}$ and $S_{d(i+1)}$, respectively.

1.7 Take $P_{c(i+1)}$ and $P_{d(i+1)}$ as the output.

4.2.2 Proposed process of motion-activity concealment

The proposed privacy protection process for motion-activity concealment is described in this section. An illustration is shown in Figure 4.3.

First of all, we detect motion parts in the image, decide a corresponding protected region R automatically, and define the video content in R as the concerned privacy-sensitive image part. Next, we select the background image part in the first image frame by the above-described method to suit the currently-processed surveillance image frame. Then, we apply the concealment process described in Chapter 3 to the privacy-sensitive image part to produce a camouflage image. In this process, we embed as well the information used to generate the background image part and the start and end positions of the detected protected region R into the resulting camouflage image to produce a protected image frame. When all image frames in the video are processed, a privacy-protected video is obtained. A detailed algorithm for this process is described in the following.

Algorithm 4.2: process for privacy-sensitive motion-activity concealment.

Input: two surveillance image sequences — a surveillance color image sequence $S_c = \{S_{c0}, S_{c1}, S_{c2}, \ldots, S_{cn}\}$ and a surveillance depth image sequence $S_d = \{S_{d0}, S_{d1}, S_{d2}, \ldots, S_{dn}\}$; a pre-selected background color image B_c, a pre-selected background depth image B_d, both with the full size of the image frame; and a secret key K.

Output: a protected color image sequence S_c' and a protected depth image sequence S_d' with the privacy-sensitive information being concealed.

Steps:

Stage 1 --- Selecting the initial privacy-sensitive image

Step 1. Find manually a rectangular-shaped region in each of the two initial input image frames, S_{c0} and S_{d0}, in which the object (a human) appears completely, consider the selected parts as part of learning results, and regard them as the initial privacy-sensitive images P_{c0} and P_{d0}, respectively.

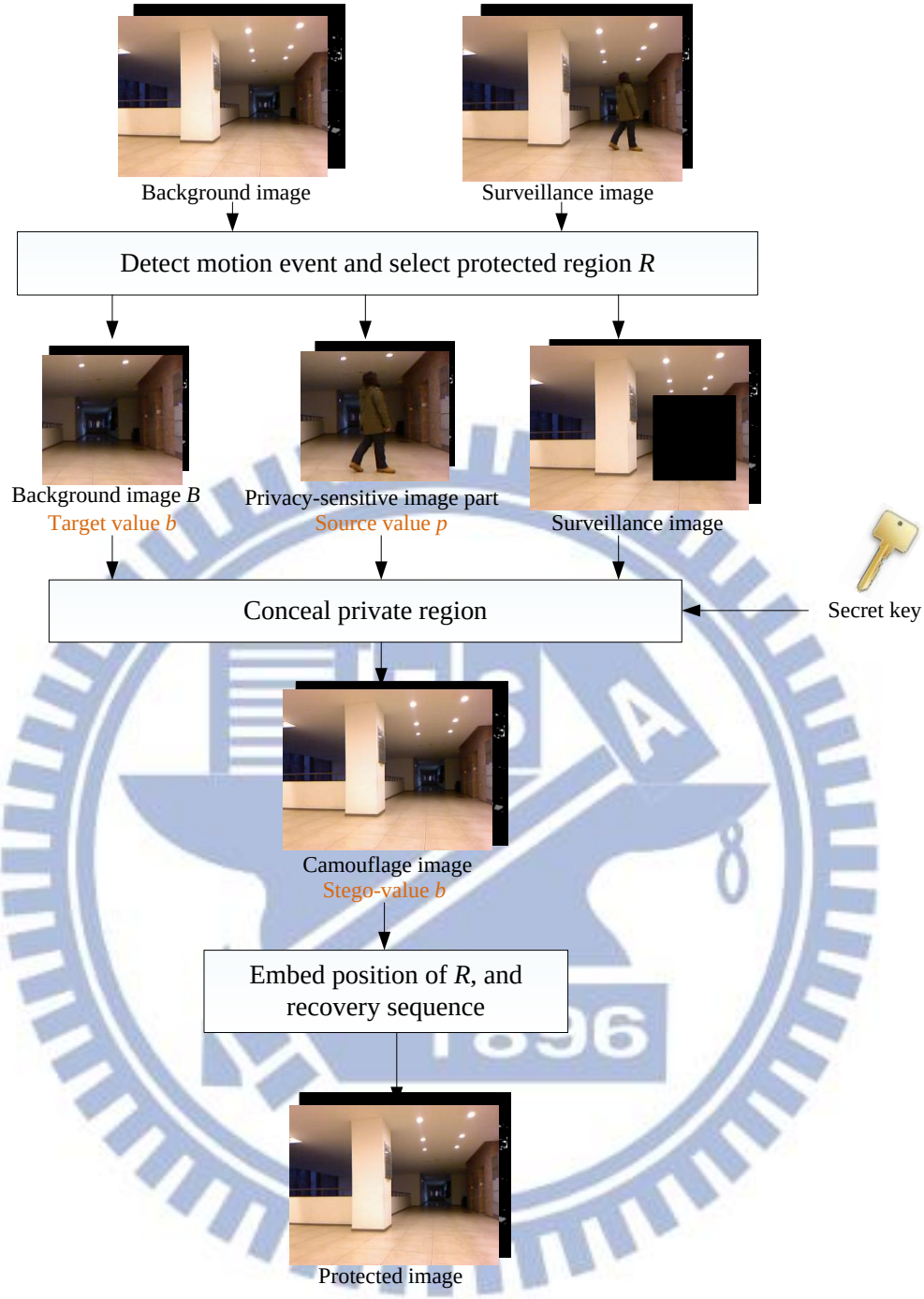


Figure 4.3 A flowchart of the proposed motion-activity concealment process.

Stage 2 --- Producing the protected surveillance image.

Step 2. With the surveillance color image sequence $S_c = \{S_{c1}, S_{c2}, \ldots, S_{cn}\}$ and the surveillance depth image sequence $S_d = \{S_{d1}, S_{d2}, \ldots, S_{dn}\}$ and the initial privacy-sensitive images P_{c0} and P_{d0} as inputs, perform Algorithm 4.1 to

detect the motion activities and decide the protected region R_i of privacy-sensitive image P_c and P_d in each pair of image frames S_{ci} and S_{di}, respectively, where $i = 1, 2, \ldots, n$.

Step 3. For $i = 1, 2, \ldots, n$, perform the following steps to conceal the image contents of S_{ci} and S_{di} in the protected region R_i:

3.1 according to the location of R_i, cut the corresponding parts of B_c and B_d out for use as the background image parts B_{ci}' and B_{di}', respectively;

3.2 with B_{ci}' and B_{di}' as input, performing Steps 1 through 7 of Algorithm 3.6 with the key K to produce a camouflage image pair S_{ci}' and S_{di}'.

3.3 Perform Steps 8 and 9 of Algorithm 3.6 to embed the lengths of the recovery sequences and the sequences themselves, which are generated in the last step (Step 3.2), into S_{ci}' and S_{di}'.

Step 4. Output the resulting image sequences S_c' and S_d' as the desired result.

4.2.3 Proposed process of motion-activity recovery

In this section, we describe how we retrieve the original privacy-sensitive image sequence from the protected image sequence. Before recovering each privacy-sensitive image frame in this process, we have to extract the recovery information from the corresponding protected image frame such as the start and end positions of the protected region, and regain the background image part as described in Section 4.2.1. With the background image part and the protected camouflage image, we can retrieve the original privacy-sensitive image content by the recovery process described in Algorithm 3.7. The above-described steps are illustrated in Figure 4.4. The details are described as an algorithm in the following.

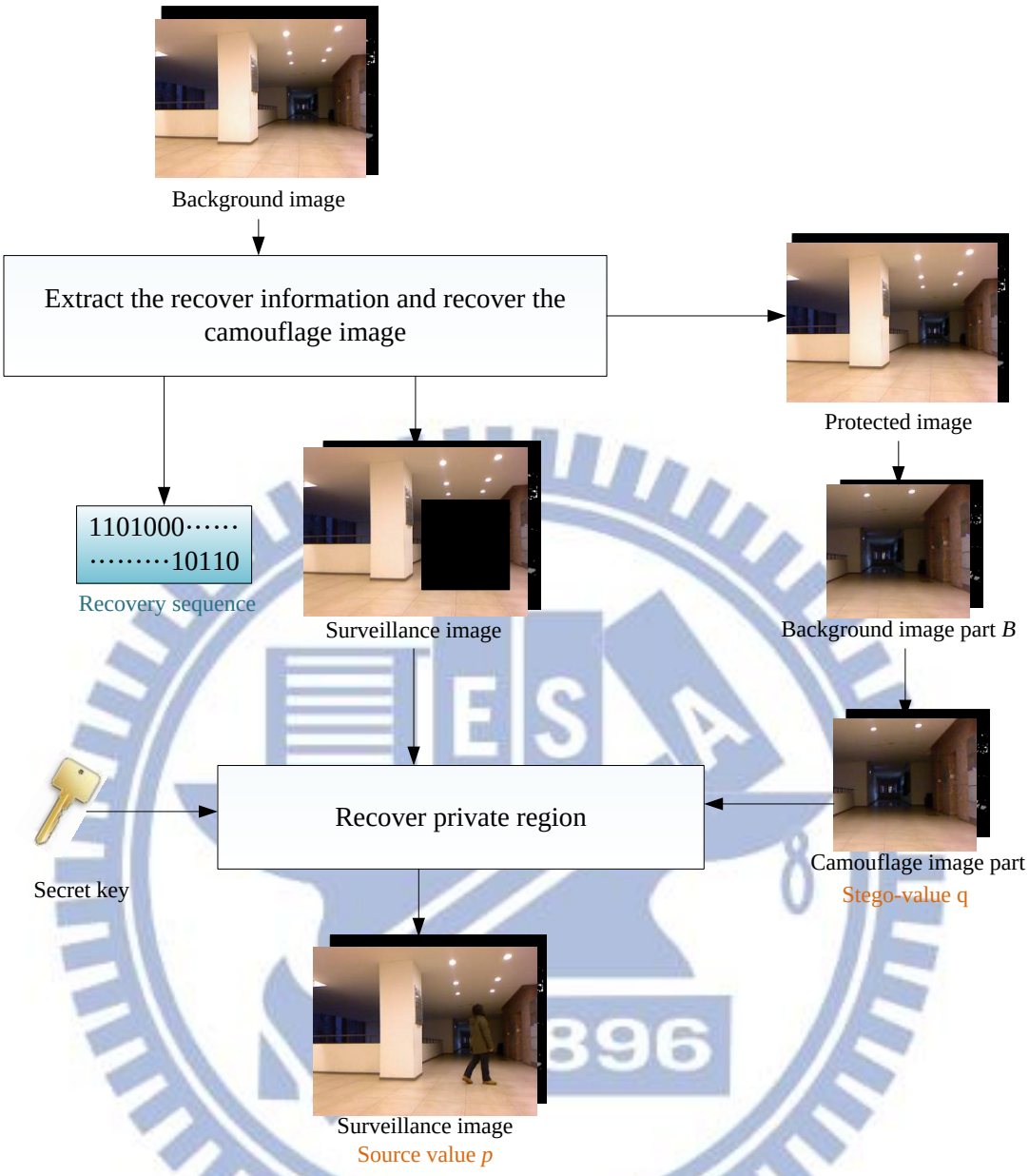


Figure 4.4 Flowchart of the private motion-activity recovery process.

Algorithm 4.3: process for the private motion-activity recovery.

Input: a protected color image sequence $S_c' = \{S_{c1}', S_{c2}', \ldots, S_{cn}'\}$ and a protected depth image sequence surveillance $S_d' = \{S_{d1}', S_{d2}', \ldots, S_{dn}'\}$; a pre-selected background color-image frame B_c, a pre-selected background depth-image frame B_d, both of the full size of the image frame; and the secret key K used in Algorithm 4.2.

Output: the original surveillance image sequence S_c and S_d recovered from S_c'' and S_d''.

Steps:

Step 1. For i = 1, 2, …, n, perform the following steps.

1.1 Extract the protected region R_i from the image contents of S_{ci}'' and S_{di}'', respectively, and retrieve the recovery sequence L_{Ri} and its length for recovering the original image contents S_c and S_d, by performing Steps 1 through 3 of Algorithm 3.7.

1.2 Cut out respectively the regions in the background images B_c and B_d corresponding to R_i as B_{ci}' and B_{di}' for use as background image parts for S_{ci}' and S_{di}', respectively.

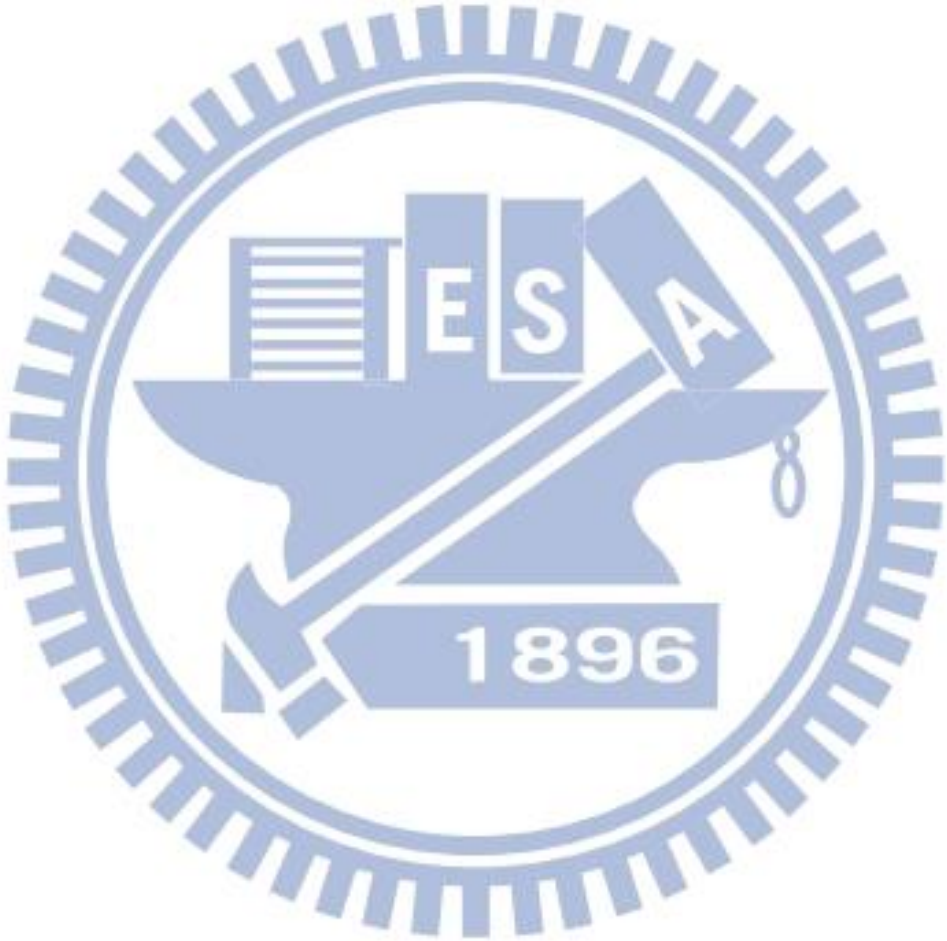
1.3 Take S_{ci}' and S_{di}' as the camouflage images and B_{ci}' and B_{di}' as the original background image parts, and perform Steps 4 through 8 of Algorithm 3.7 with the key K to recover the original privacy-sensitive images S_{ci} and S_{di}..

Step 2. Output the resulting surveillance image sequences S_c and S_d.

4.3 Experimental Results

Some experimental results of applying the proposed method for protecting privacy-sensitive motion activities in a surveillance video are shown in Figures 4.5 through 4.13. In the surveillance video, we hope that the personal movement is not leaked. First, six representative images of the protected video yielded by the proposed method using Algorithms 4.2 and 3.6 are shown in Figures 4.5 through 4.7. Next, six representative images of the recovered video yielded by the proposed method using Algorithm 4.3 are shown in Figures 4.8 through 4.10. At last, we used Algorithm 3.1 to merge the color and depth images to display the 3D surveillance video. Some frames of the results are shown in Figures 4.11 through 4.13. These experimental

results show that the information of the privacy-sensitive motion activity can be protected and recovered automatically and successfully by the proposed method.



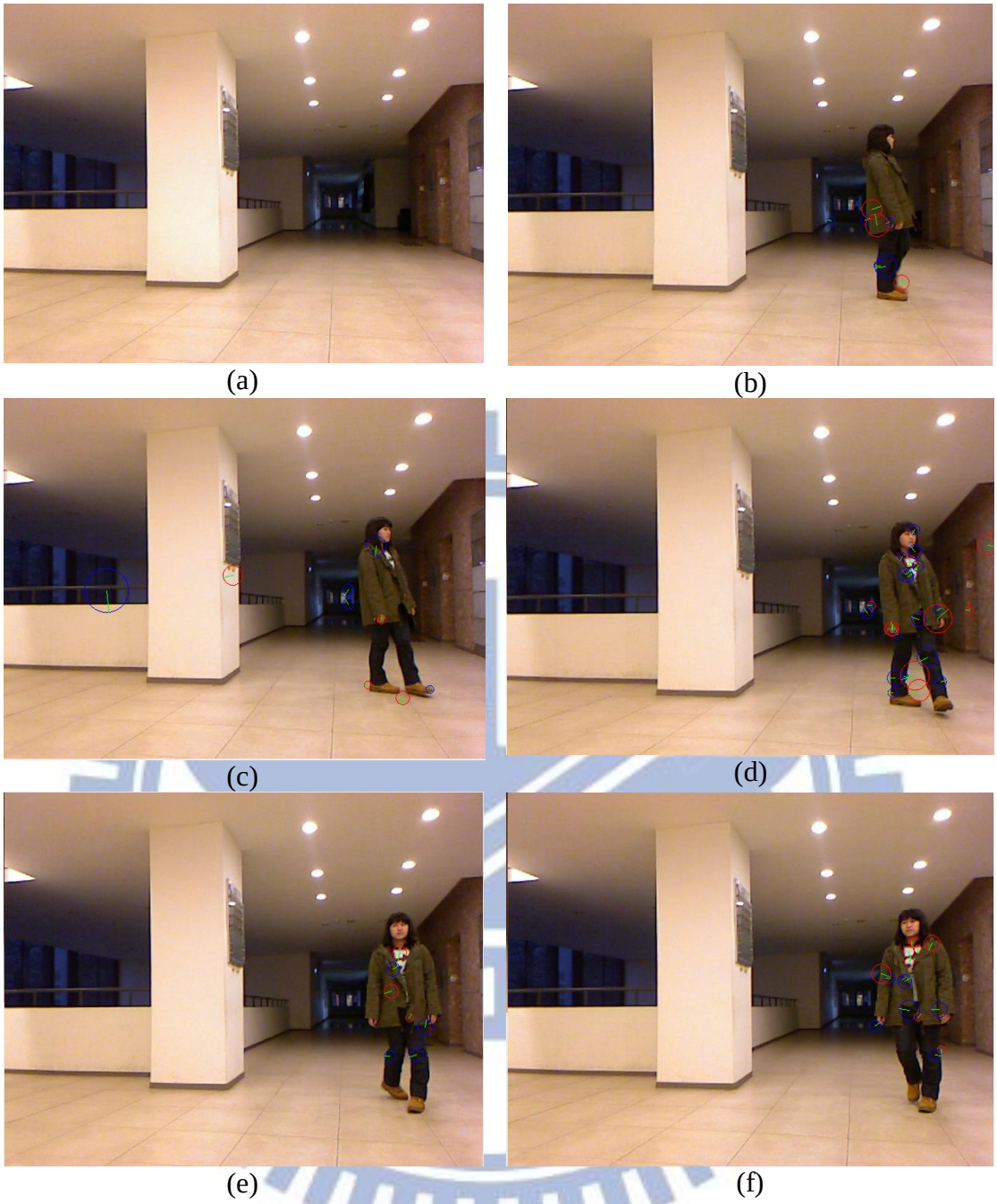


Figure 4.5 Six representative frames of a color surveillance video. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

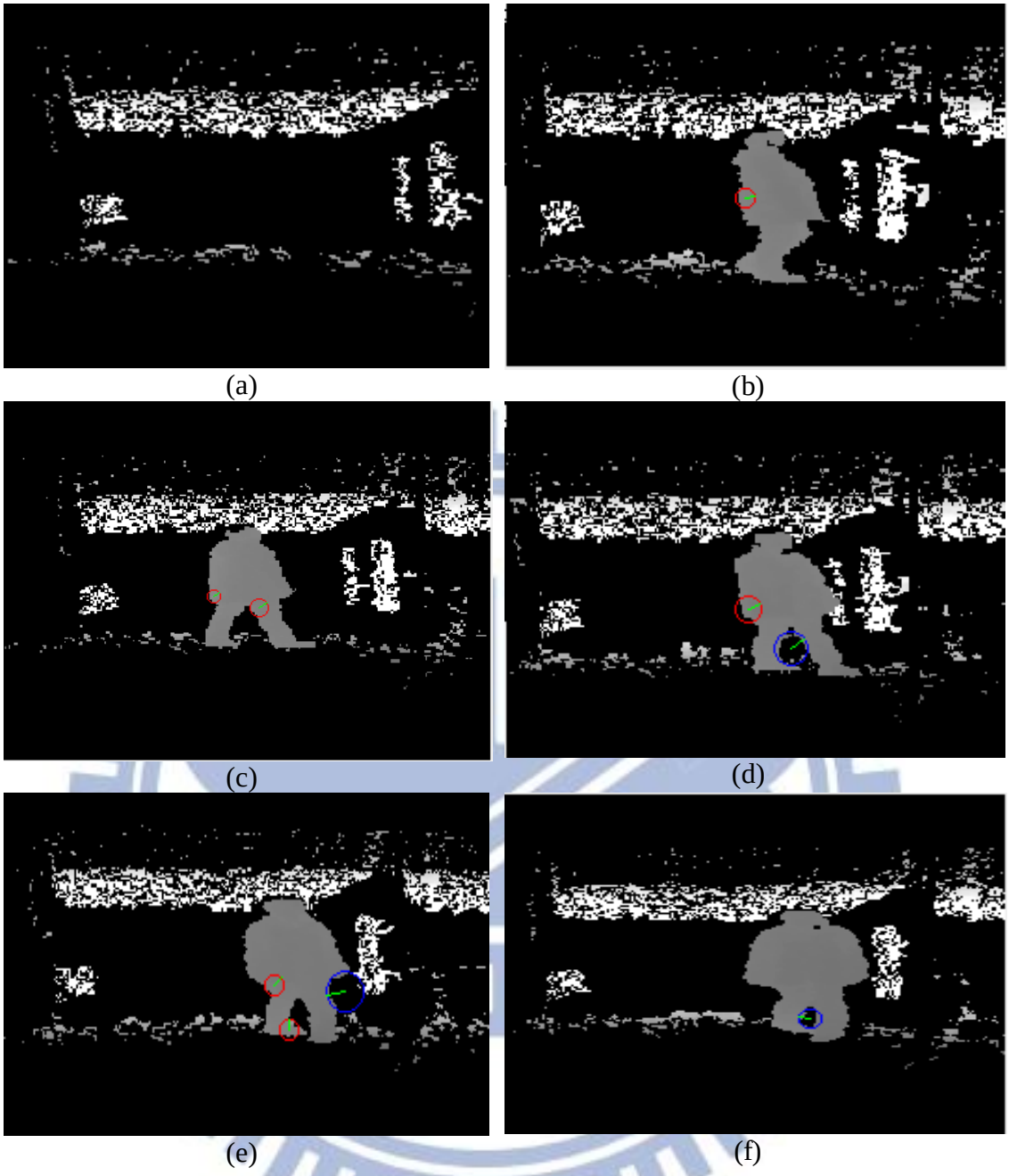


Figure 4.6 Six representative frames of a depth surveillance video corresponding to that shown Figure 4.5. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

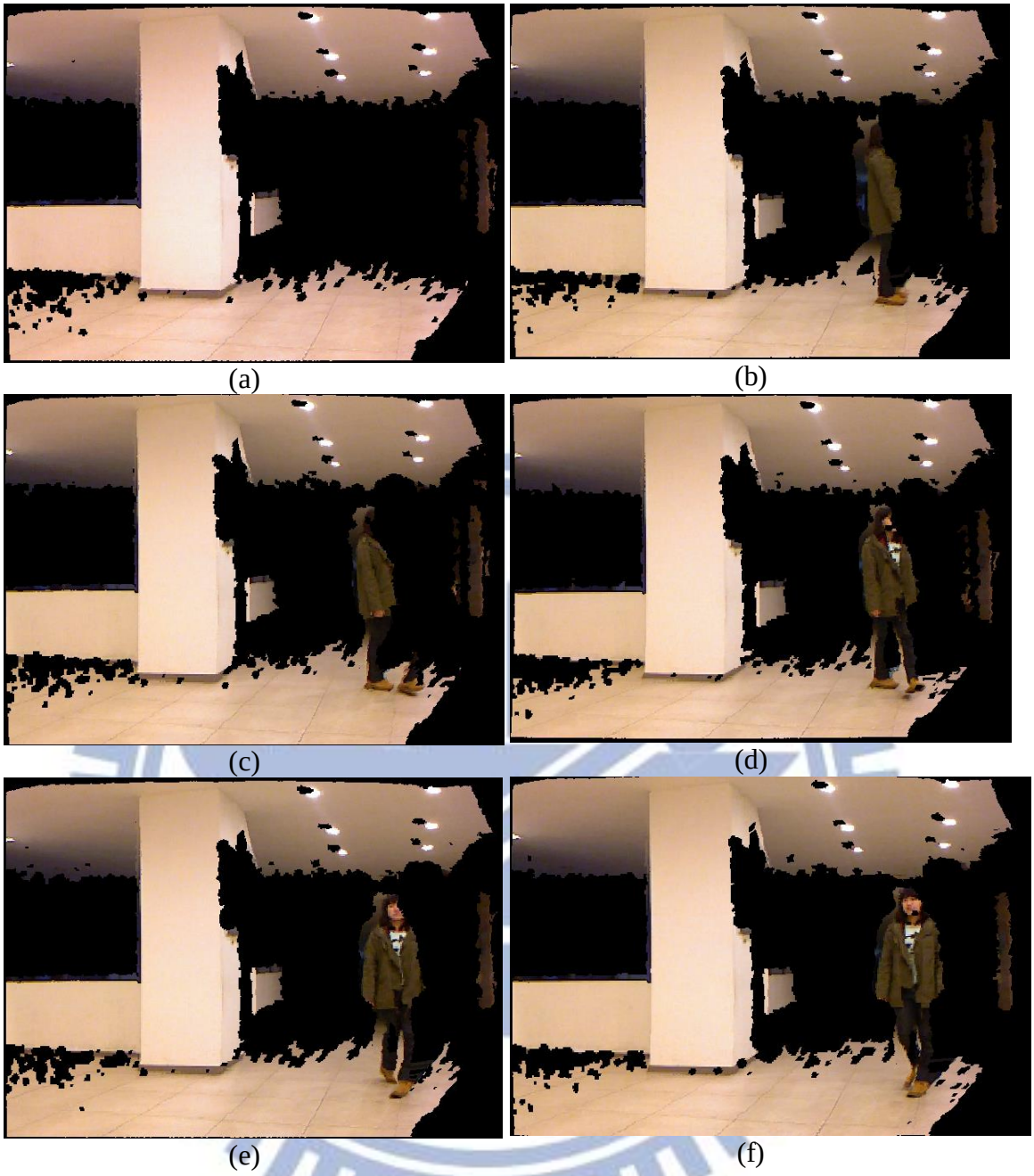
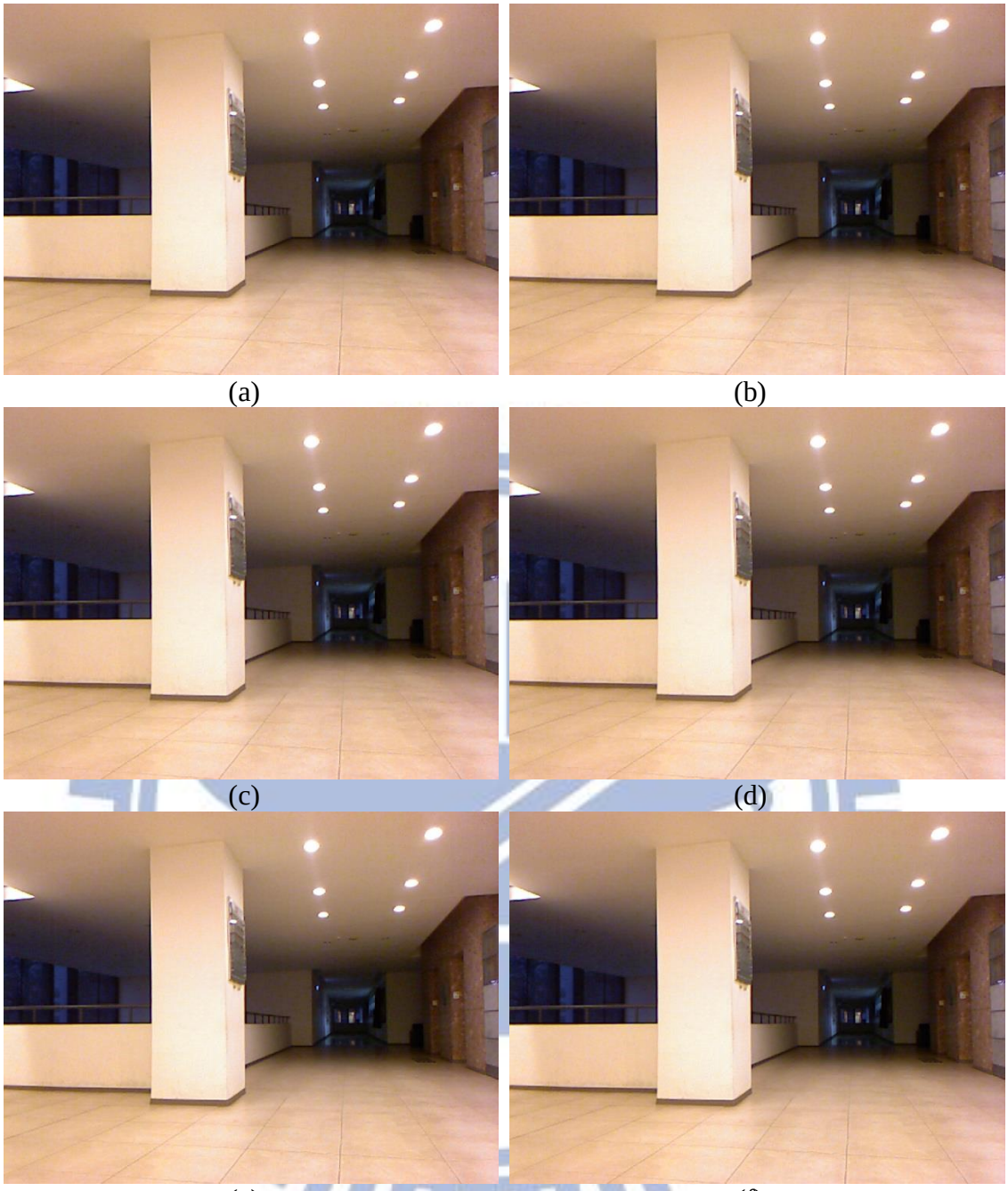


Figure 4.7 Six representative frames of a 3D surveillance video which is the result of combining those of Figures 4.5 and 4.6. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.



(a) (b)

(c) (d)

(e) (f)

Figure 4.8 Six representative frames of the privacy-protected color video yielded by the proposed method with Figures 4.6 and 4.6 as inputs. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

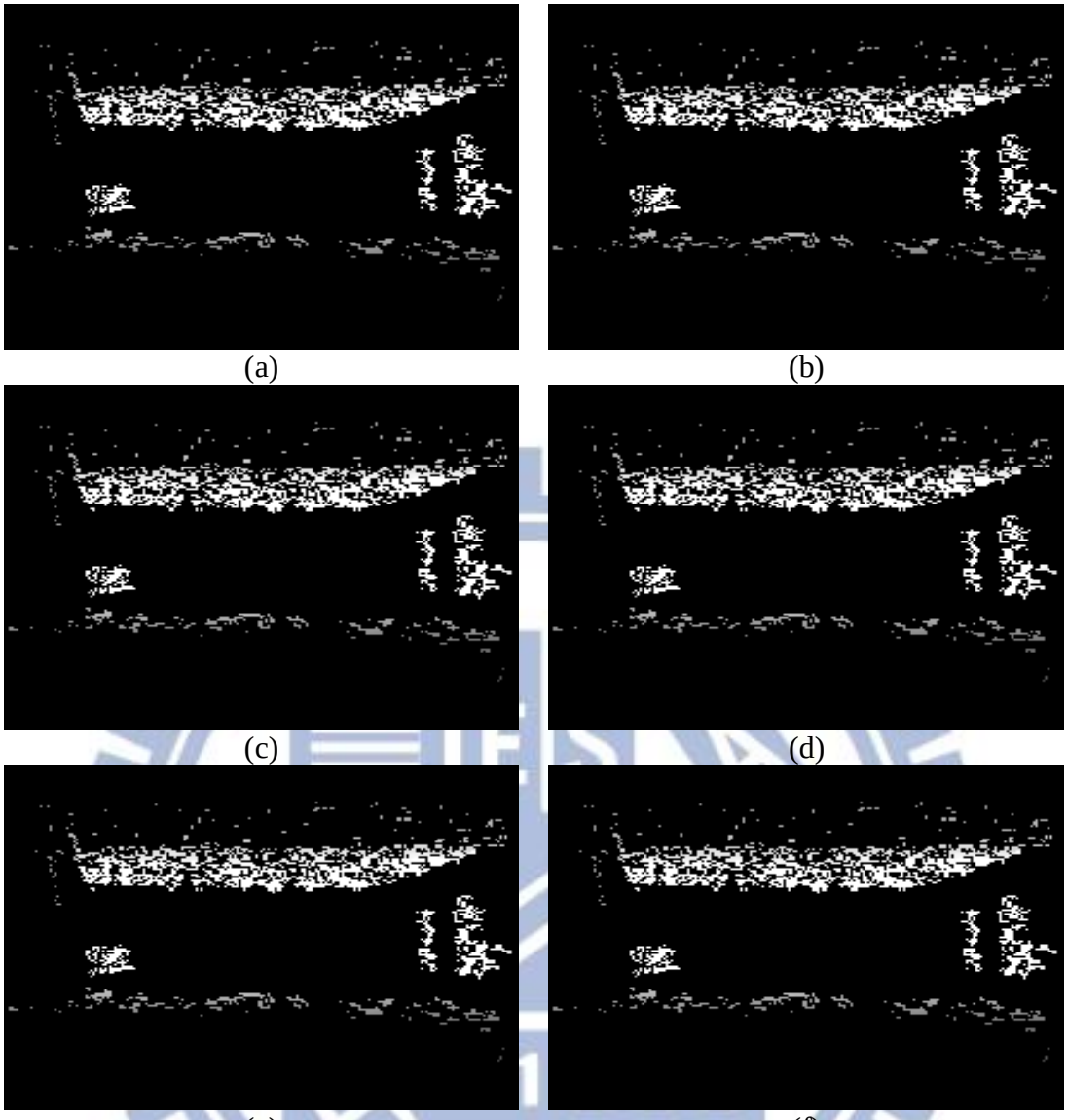


Figure 4.9 Six representative frames of the privacy-protected depth video corresponding to that of Figure 4.8. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

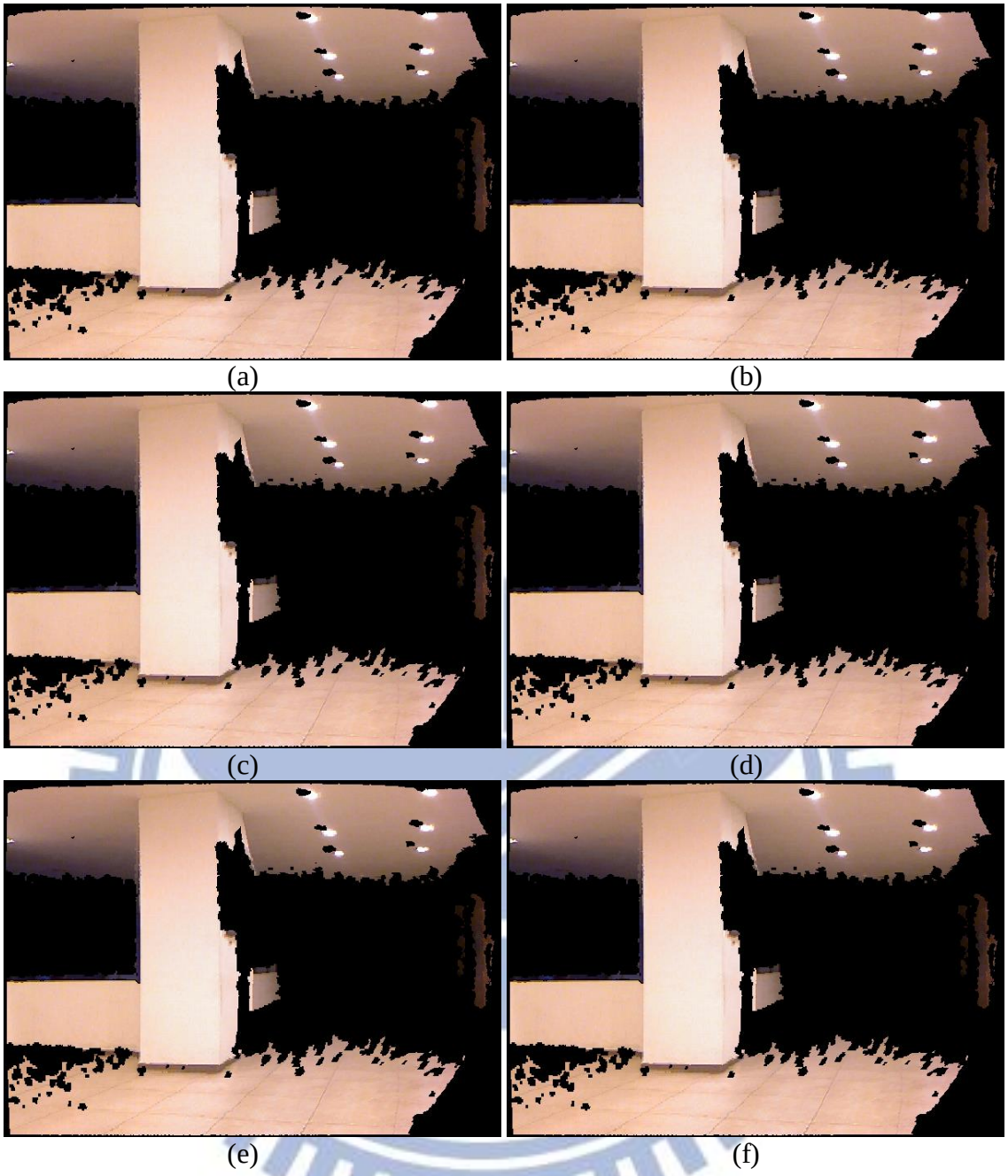


Figure 4.10 Six representative frames of the 3D privacy-protected video which comes from combination of Figures 4.8 and 4.9. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

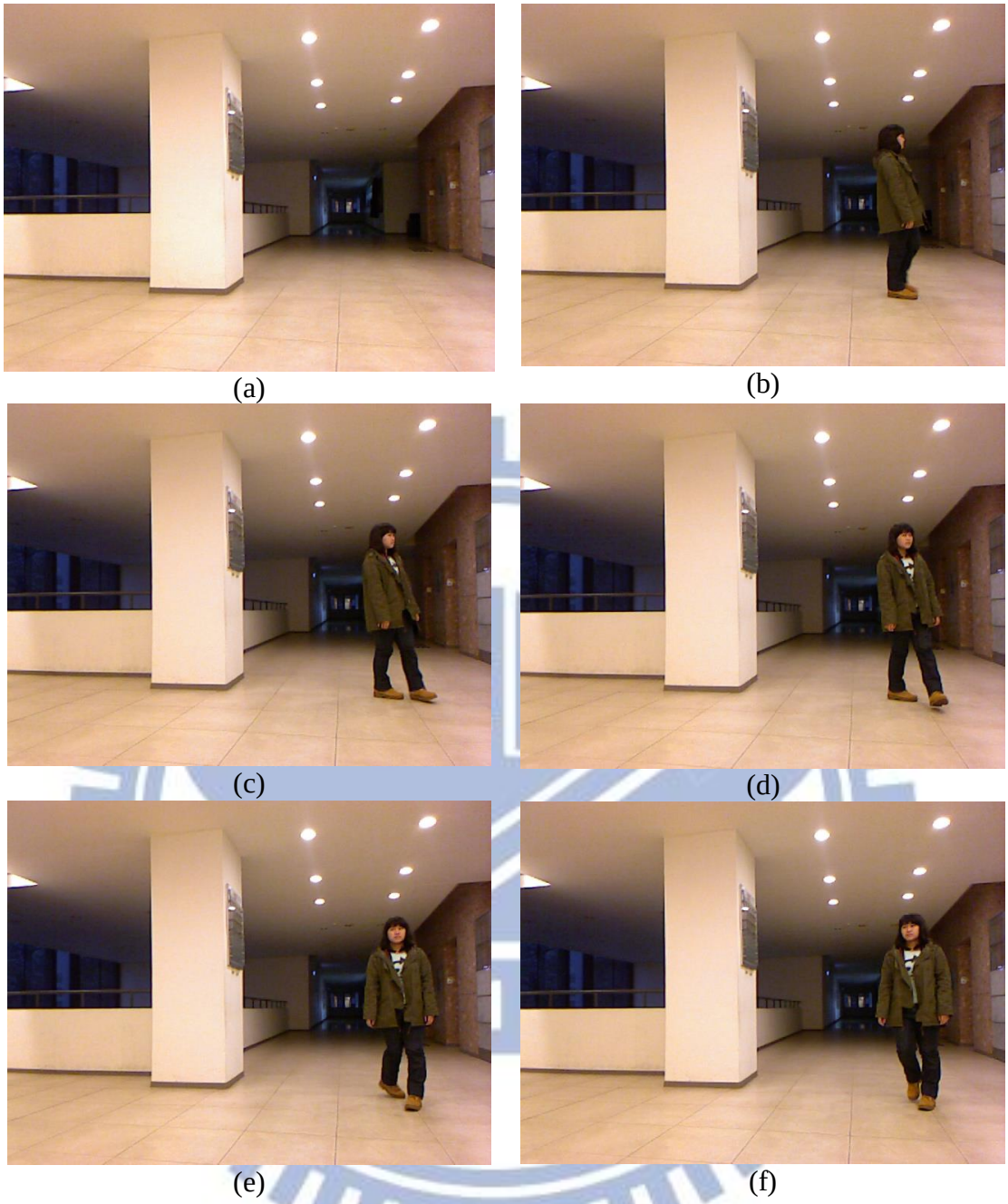


Figure 4.11 Six representative frames of the recovered color video resulting from Figure 4.8. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

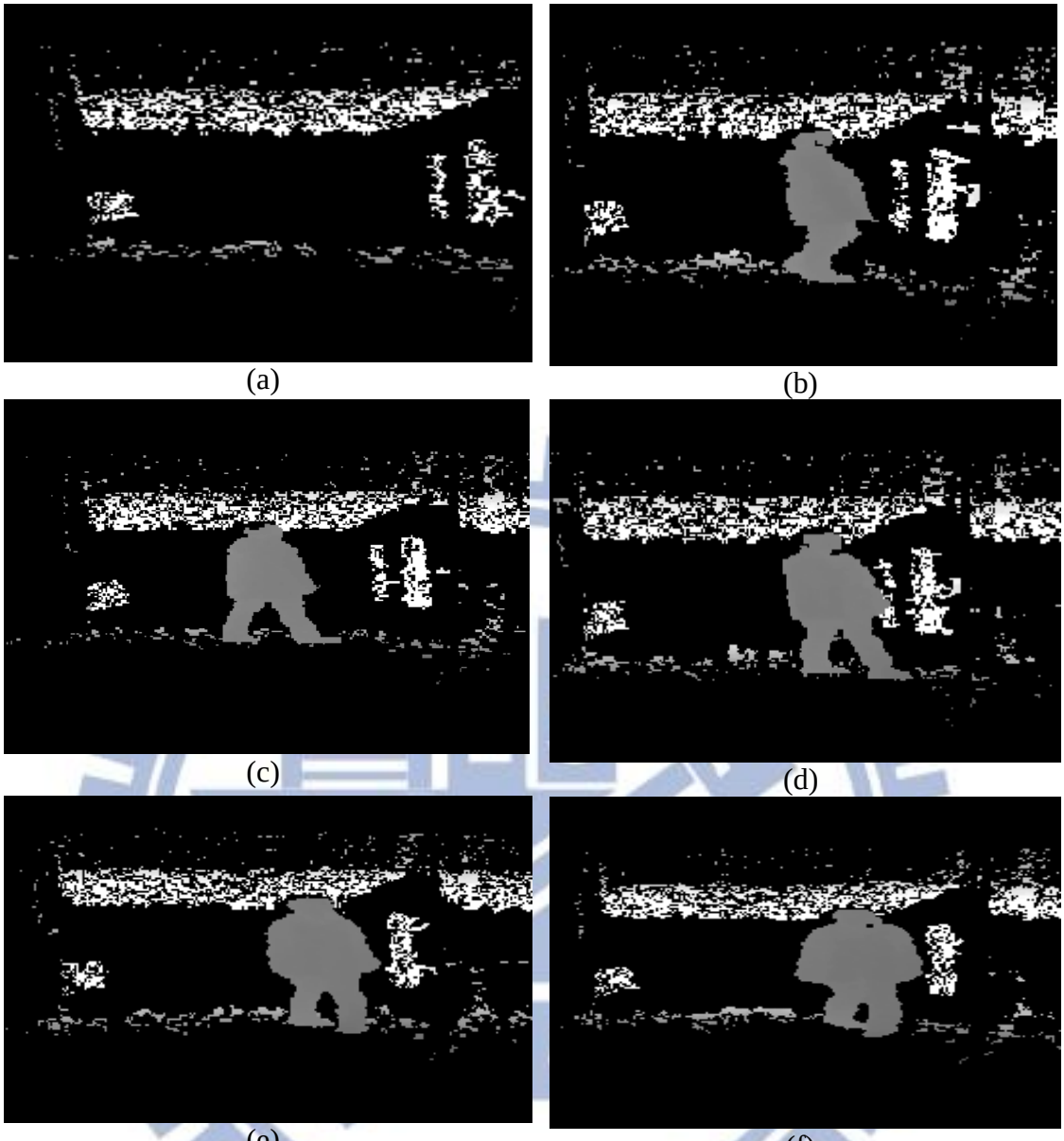


Figure 4.12 Six representative frames of the recovered depth video resulting from Figure 4.9. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

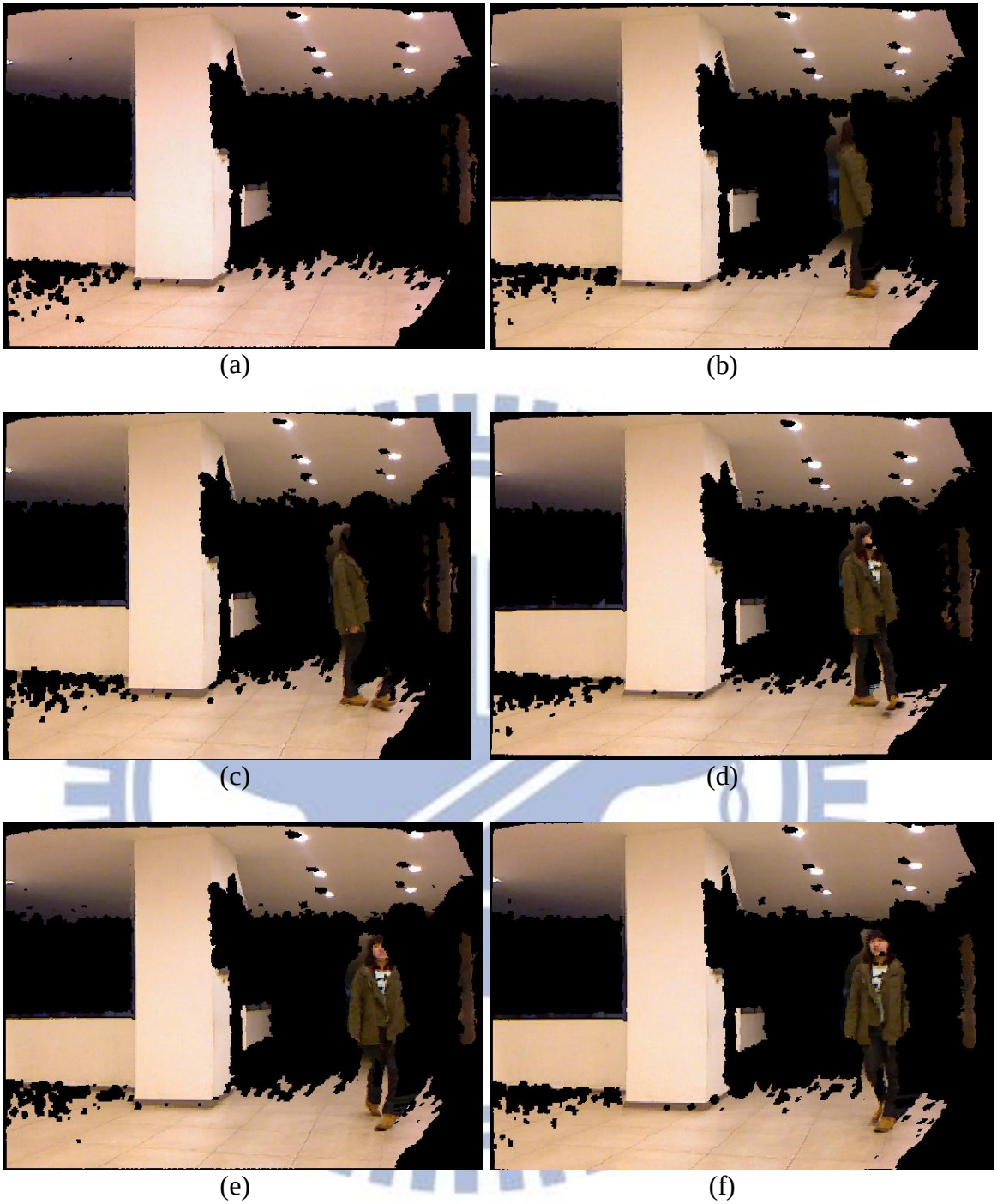


Figure 4.13 Six representative frames of the 3D recovered video combining the previously-shown color and depth images of Figures 4.11 and 4.12. (a) The background image frame. (b) The 19th frame. (c) The 20th frame. (d) The 21th frame. (e) The 22th frame. (f) The 23th frame.

Chapter 5
3D Steganography via KINECT Images

5.1 Introduction

In recent years, the various information of intellectual property becomes more and more important. Stealing the ideas of innovative creations or plagiarizing others' works is also a kind of copyright's violation. Even though such behaviors are difficult to define and prevent, image steganography is a feasible strategy to avoid such events. Due to the popularity of the Internet, image data are shared frequently on the Internet. Therefore, it is also desired to propose a method for 3D image steganography, by which a user can send secret data to other persons via the Internet or keep them securely in any digital storage.

Besides, the popularity of the 3D information which can be constructed by the KINECT device is growing dramatically. In order to reach the goal of hiding information in 3D image, we propose a method for 3D image steganography utilizing 3D images processing techniques.

In Sections 5.1.1 and 5.1.2, the related problem definitions and the idea of the proposed method are given. The proposed method for 3D image steganography by a LSB-modification scheme is presented in Section 5.2. Finally, some experimental results showing the method are given in Section 5.3.

5.1.1 Problem definition

The KINECT device has become more and more popular in recent years. Therefore, the numbers of related applications are also growing in many fields. What's more is that we can not only obtain the color and depth information, but also the three-dimensional information around the environment through the KINECT device. For example, we can use the KINECT device to capture a color image and a depth image, and then transfer the depth image into the 3D coordinates for various digital applications. The acquisition of the 3D coordinates of objects is more easily than before. Image steganography of these kinds of information will become more important.

For the reason above, we want to embed a secret image into a 3D cover image with the secret key to protect the information in this study. However, such embedding might be discovered by people. One way to deal with this issue is to hide the secret image into the *unused points* in the 3D cover image, and add the recovery information into the resulting 3D camouflage image to generate the 3D image steganography. Later, we can extract this information from the 3D camouflage image to recover the 3D secret image. The recovery information can be acquired only by an authorized user who has a secret key. That is, the user who has the secret key can remove the secret image from the 3D camouflage image and recover the embedded recovery information simultaneously. These conditions should be considered seriously when designing a method for protection of 3D image.

5.1.2 Proposed Idea

The basic idea of the proposed method for 3D steganography via KINECT images were represented in this section. At first, we combine the color image and the depth image together to form a 3D image. Then, we modify the pixel values of a given 3D cover image according to certain color and coordinate tables which are formatted as (*x*, *y*, *z*, *r*, *g*, *b*). Next, we not only embed the secret image into the cover image, but also embed the secret key to produce a camouflage image. In addition, only authorized users who have the secret key can extract the recovery information from the 3D image.

On the other hand, The original 3D image data we want to recover should be saved, and the recovery information for this purpose is created, called a *recovery sequence*. Then, we transform the resulting camouflage image into color and coordinate tables. Finally, the recovery sequence is embedded into the resulting camouflage image by a LSB-modification scheme. With this recovery sequence, the modified cover image part during the data embedding process as well as the hidden secret image can both be retrieved losslessly.

Detailed algorithms describing the proposed method and the related processes of 3D steganography are presented in the following sections.

5.2 Proposed Method for 3D Steganography via KINECT Images

In this section, the details of the proposed method for embedding the secret image into the camouflage image by a LSB-modification scheme are described. The detailed process of preprocessing of the secret image before data hiding is described

in Section 5.2.1. In Section 5.2.2, the process of secret image hiding is described. And in Section 5.2.3, the process of secret image recovery is described.

5.2.1 Preprocessing of secret image before data hiding

The idea of preprocessing the secret image before data hiding is based on Ma and Tsai [34]. The proposed method aims to transform a group of 3D images into a 3D model where the 3D images are constructed from the color and depth images acquired with a so-called octagonal 9-KINECT imaging device. The device, which consists of nine KINECT devices, can scan objects or an environment from top to bottom in a raster scan order, so that it can use the difference of height values to filter out the floor from each acquired depth image.

In addition, the unchanging nature of the static environment facilitates us to find out the position of the object in the acquired KINECT images. Therefore, in constructing an object model, we can remove undesired surrounding scene parts by eliminating those pixels with larger depth values. For example, we have used a toy bear as the object. In the recording procedure, the bear was pushed forward slowly by a person who keeps a fixed distance with respect to the bear. The fixed distance enables us to segment out the bear easier. Subsequently, we superimpose the bear appearing in every two frames at a time by use of a so-called distance-weighted correlation (DWC) measure to get a complete 3D image of the bear. In order not to choose duplicate points, we integrate all the points of the bear together without repeating identical points. An in the resulting point group, we choose every three neighboring points to form a surface, resulting in a model of the bear finally.

Now, we briefly review the measure of DWC which was proposed by Fan and Tsai [35] originally for automatic Chinese seal identification. The measure is defined as the minimum distance between two groups S and T of pixels of seal imprint images after the seal imprint images are overlapped. For each pixel p in S, a pixel in T with the minimum distance to p is searched for. If the result is a pixel p' within a limited circular area A with a pre-selected radius K, then a weight $w_p^K = 1/(d_p^2 + 1)$ is defined where d_p is the distance from p to p'; otherwise, the weight w_p^K is defined to be zero. That is, for each pixel p in S, a weight w_p^K is defined as follows:

$$\begin{aligned} w_p^K &= \frac{1}{d_p^2+1}, && \text{if } 0 \le d_p \le K, \\ w_p^K &= 0 && \text{otherwise,} \end{aligned} \tag{5.1}$$

where d_p is the distance of p in S to the closest point p' in T. Note that K is a threshold used to decide an effective distance; distances larger than this threshold are discarded. Finally, the DWC defined for the two groups of pixels, S and T, is defined as follows:

$$C^K(S,T) = \frac{1}{2} \times \left(\frac{1}{N_S} \sum_{s \in S} w_s^K + \frac{1}{N_T} \sum_{t \in T} w_t^K \right) \tag{5.2}$$

where the coefficient 1/2 is included to treat S and T symmetrically; and N_S and N_T are the total numbers of pixels in S and T, respectively. It can be verified that $0 \le C^K \le 1$ and $C^K = 1$ if and only if $S = T$. The DWC, though defined originally for seal identification, is a general measure for point-type object shape matching.

In the method proposed in this study, an extension of the above-reviewed 2D DWC measure, called 3D DWC, is used for the purpose of 3D model construction used in constructing the 3D secret message which is an object. The 2D DWC was used to decide whether the images of two object shapes are similar or not, while the 3D DWC is used to compute the displacement of two similar objects.

5.2.2 Review of the reversible contrast mapping (RCM) for lossless data hiding

We use the RCM technique proposed by Coltuc and Chassery [32] for data hiding in the proposed 3D steganography, which we review here.

For a pair of pixel values, (x_1, x_2), three cases should be handled: (a) if $0 \le 2x_1 - x_2 \le 255$ and $0 \le 2x_2 - x_1 \le 255$, 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 $0 \le 2x_1 - x_2 \le 255$ and $0 \le 2x_2 - x_1 \le 255$, 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) do not satisfy both constraints of $0 \le 2x_1 - x_2 \le 255$ and $0 \le 2x_2 - x_1 \le 255$, then we 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.

5.2.3 Proposed secret image hiding process

The proposed secret image embedding process for 3D image steganography by a LSB-modification scheme is described in this section. In the embedding process, we first transform a group of 3D images into a single 3D model where the 3D images are constructed from the color and depth images acquired with a so-called octagonal 9-KINECT imaging device. As mentioned before, we use the distance-weighted correlation (DWC) measure to get this complete 3D model of an object. Consequently, we format the coordinates of the 3D model as (x, y, z, r, g, b).

In addition, we also formulate the information of the secret message S in a 3D fashion as (x, y, z, r, g, b), such that we can embed the secret message S into a cover image C in a easier way. Then, we transform the data, add a recovery sequence, and

use a secret key K to randomize the order of the bits in the recovery sequence.

In more detail, during the data hiding process, we hide the secret image S into the unused points in the cover image C. Next, we mark each point with a label to record whether the points of the cover image C is changed or not. After these processes, the recovery information for this purpose is created, called a recovery sequence. Finally, the recovery sequence which records the label of each point is embedded into the camouflage image by a lossless LSB-modification scheme called RCM (Reversible contrast mapping) [32], and then a stego image is finally generated.

A flowchart of the proposed secret image embedding process is given in Figure 5.1 and an algorithm describing the process is given below as Algorithm 5.1. We assume that the 3D secret image S is an object or a group of points within a limited 3D space range so that it can be enclosed in the space of a cube with a certain size. Also, we assume that there exist groups of non-object points forming "holes" of cubic shapes which are large enough to contain the 3D secret image.

Algorithm 5.1: secret image embedding for image steganography.

Input: a 3D secret image S and a 3D cover image C with assumptions described above; and a secret key K.

Output: a 3D stego-image V into which S is embedded.

Steps.

Stage 1 --- Search of a "hole" in the 3D cover image larger enough to contain the 3D secret image.

Step 1. Conduct the following steps to estimate the size of the 3D secret image S.

1.1 Make a cube B_s with size $r_s \times r_s \times r_s$.

1.2 Check if all the points of the secret image S are enclosed in the cube: if not, increment r_s and repeat Step 1.1; else, continue.

1.3 Take the final cube B_{s0} and its size $r_{s0} \times r_{s0} \times r_{s0}$ as the *enclosing cube* of *S*.

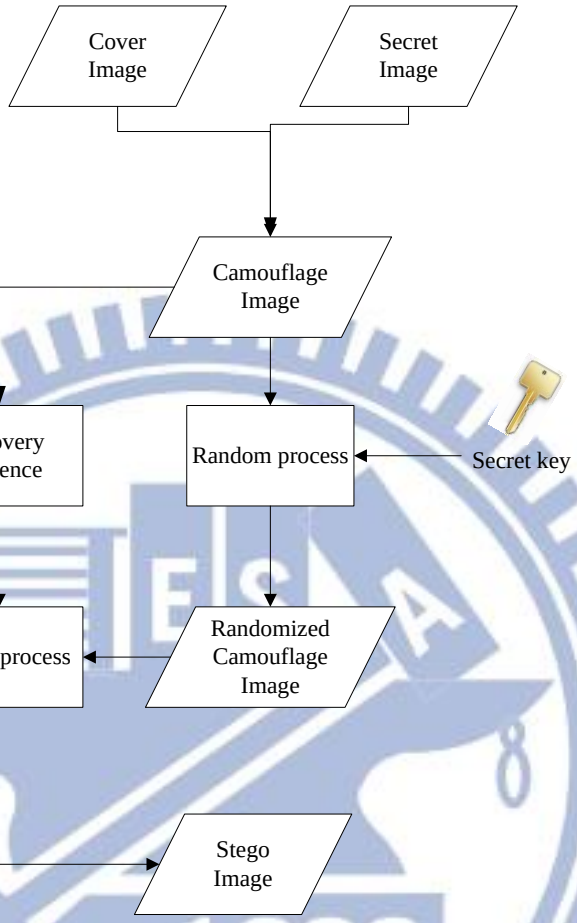


Figure 5.1 A flowchart of the proposed secret image embedding process.

Step 2. Search the 3D cover image *C* for a "hole" which is large enough to contain the 3D secret image *S* by the following steps.

2.1 Find an *unvisited non-object* point P_c in *C* in a raster scan order.

2.2 Make a cube Bc with size rc × rc × rc with Pc as the center and with the initial size set to be 3×3×3.

2.3 Check if all the points of the 3D cover image *C* within the cube are non-object points: if not, go to Step 2.1 to find another hole; otherwise, take the current cube B_c with size $r_c \times r_c \times r_c$ as a hole *H*.

2.4 Compare the size $r_{s0} \times r_{s0} \times r_{s0}$ of the enclosing cube B_{s0} of S with the size $r_c \times r_c \times r_c$ of the hole H (i.e., compare r_{s0} with r_c) to see if H can contain the cube B_{s0} of S: if so, go to Step 3; else, enlarge the hole by increment r_c by one and go to Step 2.2.

Stage 2 --- Embedding the 3D secret image into the found "hole" in the 3D cover image.

Step 3. Embed S into the current cube B_c of C with the size $r_c \times r_c \times r_c$ with the key K to produce a camouflage image C_a as follows.

3.1 Move S to the center of the cube B_c in C, which can contain S.

3.2 Record in order the side length r_c and the coordinates (x_c, y_c) of the center point P_c of the cube B_c as a recovery sequence L_R.

3.3 Randomize the positions of the pixels in S by the secret key K to produce a camouflage image C_a.

Step 4. Perform the following steps to embed the recovery sequence L_R into C_a to produce a 3D stego-image V.

4.1 Transform L_R into a binary string S_R, count the number N of bits in S_R, and convert N into a 18-bit binary string as N_R.

4.2 Combine strings N_R and S_R by adding N_R to the front of string S_R.

4.3 Embed the first three unembedded bit b_s of S_R respectively into the R, G, and B color values of a pair of *object* pixels in C_a selected a raster-scan order (with all the bits in S_R regarded as unembedded initially) by the lossless RCM scheme [32] as reviewed in Section 5.2.2.

4.4 Repeat Step 4.3 until all the bits in S_R are embedded in C_a.

Step 5. Output the resulting C_a as the 3D stego-image V.

Note that in Step 4.3 above, while carrying out the RCM scheme, if case (c) with the pair of pixel values being x_1 and x_2 is encountered as described in Section 5.2.2, the LSB of x_1 should be changed to be "0," and the original LSB of x_1 should be appended to the end of the data string S_R for hiding.

5.2.4 Proposed secret image recovery process

In the secret image recovery process, we have two input sets of data for the image retrieval work. The first is a stego-image, and the second is a user key. If the given user key is different from the one which is used in the secret image embedding process, the extraction of the secret image will fail. By using a reverse version of the RCM scheme and the right key, we can recover the original secret image from the stego-image. The detailed steps and a flowchart of the proposed secret image recovery process are given in Algorithm 5.2 and Figure 5.2, respectively.

Algorithm 5.2: secret image recovery.

Input: a 3D stego-image V and a secret key K.

Output: a recovered 3D secret image S.

Steps.

Step 1. Perform the following steps to extract the length of recovery sequence L_R from V.

1.1 Select an object pixel pair, T_1 and T_2, from V in a raster-scan order.

1.2 Extract the three LSBs respective from the R, G, and B color values of T_1 and T_2 by an inverse version of the lossless RCM scheme [32] and append them into a string N_R (initially empty).

1.3 If the length of sequence N_R is not equal to 18, then go to Step 1.1 to extract more bits to compose N_R; otherwise, continue.

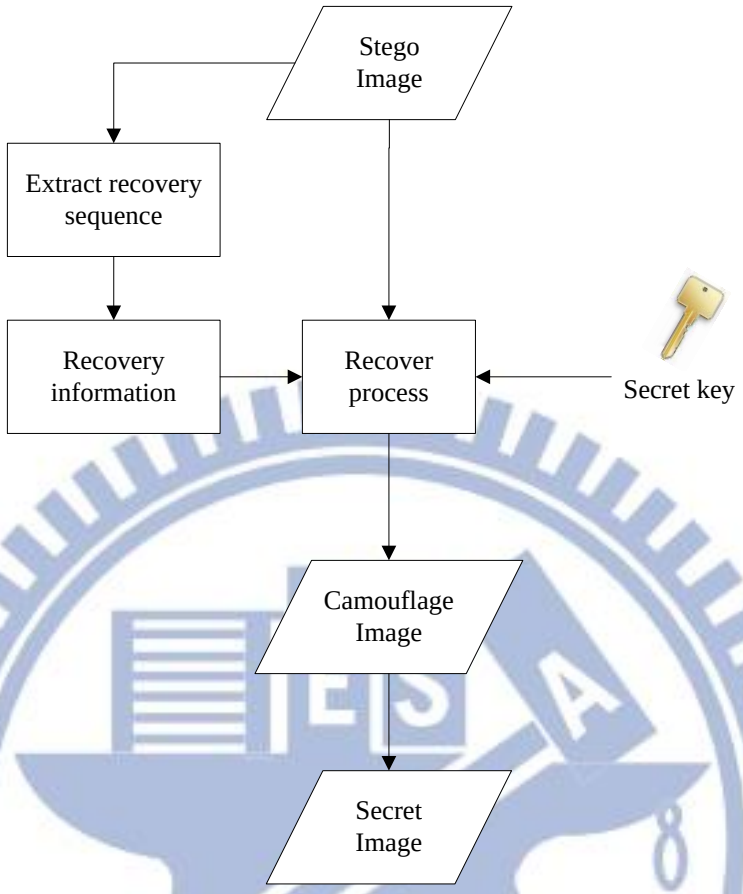


Figure 5.2 Flowchart of proposed secret image recovery process.

Step 2. Transform N_R into a decimal integer L and regard it to be the length of the recovery sequence L_R to be extracted next.

Step 3. Perform the following steps to extract the recovery sequence L_R from V.

3.1 Select an unprocessed object pixel pair T_1 and T_2 from V in a raster-scan order.

3.2 Extract three LSBs respectively from the R, G, and B color values of T_1 and T_2 by an inverse version of the lossless RCM scheme [32] and append them to a string S_R (initially empty).

3.3 If the length of string S_R is not equal to L, then go to Step 3.1;

otherwise, continue.

Step 4. Transform the extracted string S_R into the recovery information of a side length r_c and the coordinates (x_c, y_c) of the center point P_c of a cube B_c within the stego-image V.

Step 5. According to the side length r_c and the coordinates (x_c, y_c), find the cube B_c with the size $r_c \times r_c \times r_c$ and the center point at coordinates (x_c, y_c) in V, and extract all the object pixels of V within the cube B_c as V'.

Step 6. De-randomize the positions of the pixel values in V' using the secret key K, and then take the resulting image as a 3D secret image S.

Step 7. Output the 3D secret image S as the final result.

5.3 Experimental Results

Some experimental results of applying the proposed method for 3D image steganography are shown in Figures 5.3 through 5.6. Figures 5.3 and 5.4 are the 3D secret image and 3D cover image given by the user, respectively. Figure 5.5 shows some different views of the 3D stego-image generated from the secret and cover images using Algorithm 5.1.

By taking Figure 5.5 with a secret key as inputs to Algorithm 5.2, we recovered the original cover image as shown in Figure 5.6, where several views of the recovered cover image are shown.

Figure 5.3 A 3D secret image.

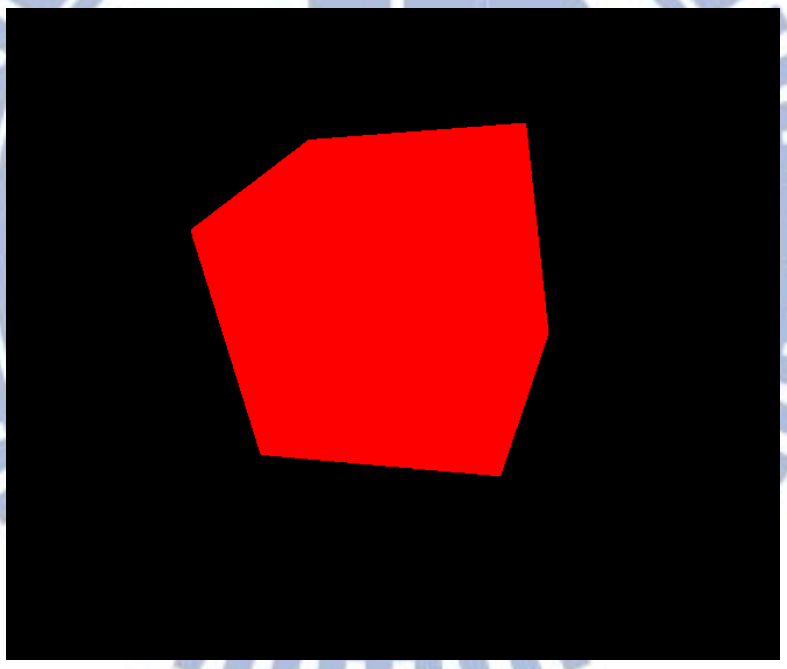


Figure 5.4 A 3D cover image.

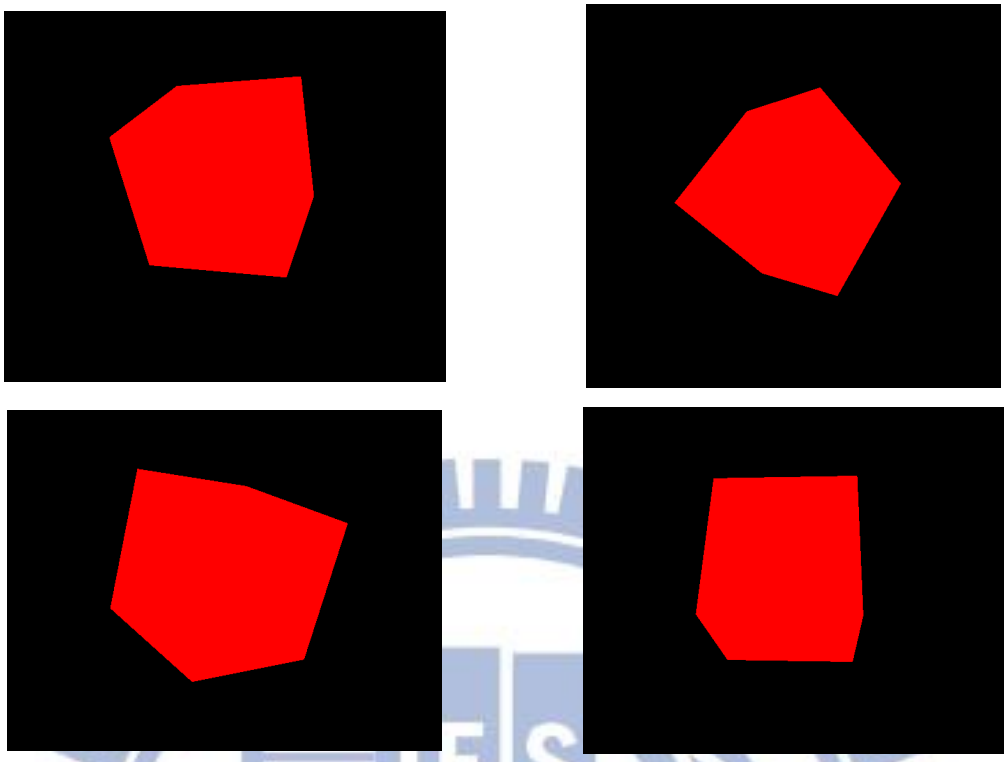


Figure 5.5 A 3D stego-image generated from Figures 5.3 and 5.4 by the proposed method seen from different viewpoints.

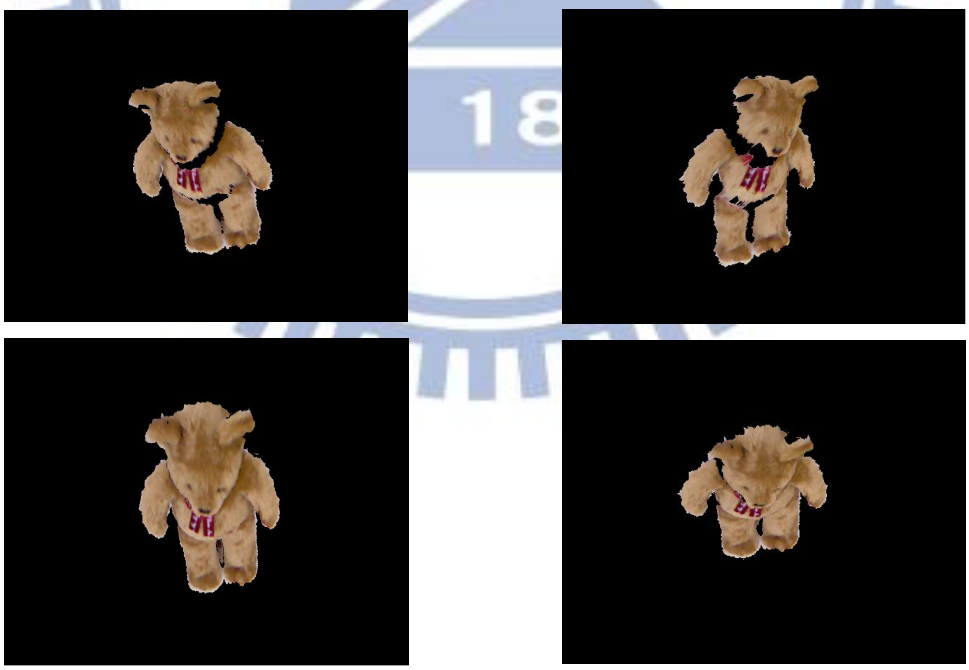


Figure 5.6 The secret image extracted from the stego-image shown in Figure 5.5 seen from different viewpoints.

Chapter 6
Conclusions and Suggestions for Future Works

6.1 Conclusions

In this study, we have proposed several methods for a variety of data hiding applications, including privacy protection of selected regions and motion activities in surveillance videos and 3D image steganography. For privacy protection of selected privacy-sensitive regions in videos, a method for concealing privacy-sensitive image part in a surveillance image frame by a prediction-based mapping scheme has been proposed. Specifically, the privacy-sensitive image part with the source pixel values is mapped into a prediction-residue image with residue values. This prediction-residue image and the background image part with target values then are taken as inputs to the inverse of the prediction-based mapping function mentioned above to yield a camouflage image with stego-values, which looks similar to the background image. The privacy-sensitive image part concealed in the camouflage image can be retrieved losslessly.

For privacy protection of motion activities, at first we segment respective privacy-sensitive image from currently-processed surveillance images automatically. Next, we extract feature points from these images using the SURF extraction algorithm and match them with a reference image extracted from the previous image frame to locate the human in the current image frame. Al last, the protected region including the located human is hidden by the above-mentioned privacy protection process.

For 3D image steganography, at first we use the distance-weighted correlation (DWC) measure to construct a complete 3D image from the color and depth images acquired by the KINECT device. Then, we hide the 3D secret image and a related a recovery sequence into a hole with no object point in the input 3D cover image according to the reversible contrast mapping (RCM) scheme to produce a 3D stego-image. In the 3D secret image recovery process, with the extracted data of the recovery sequence, we can derive the originally 3D secret image from the stego-image using a reverse version of the RCM scheme.

The experimental results shown in the previous chapters have revealed the feasibility of the proposed system.

6.2 Suggestions for Future Works

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

(1) It is interesting to modify the proposed methods for different applications using multiple KINECT devices.

(2) It is interesting to modify the proposed methods for direct uses on 3D images for video surveillance instead of on 2D color images.

(3) It is interested to apply image inpainting techniques to fill up non-object surface points to improve 3D image steganography.

(4) It is interesting to generalize the proposed image steganography method for video-type input secret data.

(5) It is desirable to apply the proposed methods to more applications, such as covert communication via images or authentication of images.

References

[1] T. Y. Liu, and W. H. Tsai, "Generic Lossless Visible Watermarking — A New Approach," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1224-1235, May 2010.

[2] J. K. Paruchuri, S. S. Cheung, and M. W. Hail, "Video data hiding for managing privacy information in surveillance systems," *EURASIP Journal on Information Security*, vol. 2009, pp.1-18, Jan 2009.

[3] E. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face image," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, Feb. 2005.

[4] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 11168-1174, Aug. 2008.

[5] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, March 2006.

[6] G. Xuan, J. Zhu, J. Chan, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *IEEE Electronics Letters*, vol. 38, no. 25, pp. 1646-1648, December 2002.

[7] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no 2, pp. 253-266, February 2005.

[8] I. J. Lee, J. Y. Min, H. Lee, S. Min, S. Kim, "A scaling parameter optimization of watermarking using autostereogram as random dot images," in *International*

Conference on Visual Information Engineering (VIE 2003), pp. 314-316, January 2003.

[9] R. Z. Wang and Y. S. Chen, "High-payload image steganography using two-way block matching," *IEEE Signal Processing Letters*, vol. 13, no. 3, pp. 161-164, Mar. 2006.

[10] Prabakaran.G and Bhavani.R, "A modified secure digital image steganography based on discrete wavelet transform", *International Computing Electronics and Electrical Technologies (ICCEET*), pp. 1096-1100, March 2012.

[11] Kinect, "Developer SDK, Toolkit & Documentation, Kinect for Windows," Available in www.microsoft.com/en-us/kinectforwindows/develop/, Accessed in May. 2013.

[12] OpenNI. Available online: http://www.openni.org/, Accessed in May. 2013.

[13] OpenKINECT. Available online: http://openkinect.org/wiki/Main_Page/, Accessed in May. 2013.

[14] Jan Smisek, Michal Jancosek and Tomas Pajdla, "3D with KINECT", *IEEE International Conference on the Computer Vision Workshops (ICCV Workshops)*, pp. 1154-1160, Nov. 2011.

[15] Leandro Cruz, Djalma Lucio, Luiz Velho, "KINECT and RGBD Images: Challenges and Applications", 2012 25th *SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, pp. 36-49, Aug 2012.

[16] Yang Zhao, Zicheng Liut, Lu Yang and Hong Cheng, "Combing RGB and Depth map features for human activity recognition", *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pp. 1-4, Dec 2012.

[17] Jaeyong Sung, Colin Ponce, Bart Selman and Ashutosh Saxena, "Unstructured human activity detection from RGBD images", *IEEE International Conference*

on Robotics and Automation (ICRA), pp. 842-849, May 2012.

[18] Ningbo Wang, Xiaojin Gong, Jilin Liu "A new depth descriptor for pedestrian detection in RGB-D images", *21st International Conference on Pattern Recognition (ICPR)*, pp. 3688-3691, Nov 2012.

[19] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," *IEEE Workshop Applications of Computer Vision, Princeto*n, NJ, USA, pp. 8-14, Oct. 1998.

[20] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 809–830, Aug. 2000.

[21] Lili Dai, Yuye Wang and Jingwen Wang, "The motion detection method apply to video surveillance embedded system", *2011 International Conference on Multimedia Technology (ICMT*), Harbin, China, pp. 5162-5165, July 2011.

[22] Nahum Kiryati, Tammy Riklin Raviv, Yan Ivanchenko and Shay Rochel, "Real-time Abnormal Motion Detection in Surveillance Video", *19th International Conference on Pattern Recognition, ICP*R 2008, pp. 1-4, Dec 2008.

[23] Hyenkyun Woo, Yoon Mo Jung, Jeong-Gyoo Kim, and Jin Keun Seo, "Environmentally Robust Motion Detection for Video Surveillance", *IEEE Transactions on Image Processing, Seoul, South*, pp. 2838-2848, Nov 2010.

[24] Yongseok Yoo and Tae-Suh Park, "A moving object detection algorithm for smart cameras", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW* '08, pp. 1-8, June 2008.

[25] T. Y. Lin, and W. H. Tsai,, "A Study on New Data Hiding Techniques Using Reversible Prediction-based Mapping for Video Surveillance and Steganography Applications", *M. S. Thesis*, Institute of Multimedia Engineering, National Chiao

Tung University, Hsinchu, Taiwan, Republic of China, June. 2011.

[26] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 809-830, Aug. 2000.

[27] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," *Proceedings of IEEE Workshop Applications of Computer Vision, Princeton, NJ, USA*, pp. 8-14, Oct. 1998.

[28] W. Zeng, J. Du, W. Gao and Q.M. Huang, "Robust moving object segmentation on H.264/AVC compressed video using the block-based MRF model," *Real-Time Imaging*, vol. 11, pp. 290-299, Aug. 2005.

[29] R.V. Babu and A. Makur, "Object-based surveillance video compression using foreground motion compensation," *IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, pp. 458-463, Dec. 2006.

[30] S. K. Kapotas and A. N. Skodras, "Moving object detection in the H.264 compressed domain," *IEEE International Conference on Imaging Systems and Techniques (IST), Thessaloniki, Greece*, pp.325-328, 1-2 July 2010.

[31] S. Bedi, E.A. Edirisinghe,G. Grecos, "Improvements to the JPEG-LS prediction scheme," *Image and Vision Computing*, vol. 22, pp. 9-14, 2004.

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

[33] Herbert Bay, Tinne Tuytelaars and Luc Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, vol. 110, Issue 3, pp. 346-359, June 2008.

[34] P. C. Ma, and W. H. Tsai,, "3D Environment Modeling and Monitoring via KINECT Images for Video Surveillance", *M. S. Thesis*, Institute of Computer

Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China, June. 2013.