國立交通大學

資訊科學與工程研究所

碩士論文

應用資訊隱藏技術做安全監控之研究

A Study on Applying Information Hiding Techniques to Security Surveillance

研 究 生：陳建中

指導教授：蔡文祥　教授

中華民國 九十六 年六月

應用資訊隱藏技術做安全監控之研究

A Study on Applying Information Hiding Techniques to Security Surveillance

研究生：陳建中　　Student: Jian-Jhong Chen

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

國立交通大學
資訊科學與工程研究所
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 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

應用資訊隱藏技術作安全監控之研究

研究生：陳建中　　　指導教授：蔡文祥 博士

國立交通大學資訊科學與工程研究所

摘要

現今，越來越多監控視訊被使用來以預防犯罪。本論文提出三種資訊隱藏技術在視訊監控系統上的應用。第一個應用是在視訊中尋找特定的人和移動物體的方法。我們抽取出視訊影像中移動部分的特徵並且將其隱藏在影像之中，之後藉此來尋找具有相同移動物體特徵的影像。第二個應用是把含有移動資訊的影像保存在視訊中較安全的片段中，以預防被惡意竄改。我們將含有移動資訊的可疑影像隱藏到背景場景中的Ｉ框架影像(I-frame)中，並將預先定義好的驗證訊號藏入其中。經由一本文所提驗證程序，在視訊中修改或置換相關的影像可被偵測出來，並且可經由還原程序將影像還原回來。最後，第三個應用在監視汽車被非法入侵的事件。在車上的電腦會將警告訊息以及車牌號碼經由無線網路傳送到監視系統，這些訊息接著會被即時隱藏到相關的視訊影像中，因此入侵事件的影像可以被檢索出來。最後我們提出了相關的實驗結果，證明了所提方法之可行性。

A Study on Applying Information Hiding Techniques to Security Surveillance

Student: Jian-Jhong Chen Advisor: Prof. Wen-Hsiang Tsai, Ph. D.

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

ABSTRACT

Nowadays, more and more surveillance videos are recorded to prevent crimes. Three methods for data hiding applications implemented on video surveillance systems are proposed. In the first method proposed for the purpose of searching videos for specified persons or motion objects, we extract features from motion parts of video frames and embed them in the frames to help later searches for frames which contain motion objects with similar features. In the second method proposed for keeping motion frames in safer video sections to prevent malicious tampering, we embed concerned motion frames together with specially-designed authentication signals into I-frames with background scenes. Modification or replacement of relevant motion frames can be checked via an authentication process and compensated by a frame recovery process. Finally, the third method is proposed to monitor car break-in events, in which a warning message of event time and the car plate number is sent wirelessly from an on-car computer to a surveillance system. The message is then embedded in relevant video frames. Such event frames can be retrieved later for inspection. Good experimental results prove the feasibility of the proposed methods.

ACKNOWLEDGEMENTS

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

Thanks are due to Mr. Chih-Jen Wu, Mr. Kuan-Chieh Chen, Miss Kuan-Ting Chen, Mr. Tsung-Chih Wang, Mr. Yi-Fan Chang, and Mr. Shang-Huang Lai for their valuable discussions, suggestions, and encouragement. 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 my thesis study.

Finally, I also extend my profound thanks to my family for their lasting love, care, and encouragement. I dedicate this dissertation to my beloved parents.

CONTENTS

LIST OF FIGURES

Chapter 1
Introduction

1.1 Motivation

With the rapid rise of crime rates nowadays, environment surveillance systems become more and more essential for security monitoring in public places. Digital videos, which are popular multimedia files to store image sequences, are often used to store the result of security monitoring. Generally speaking, most frames in such videos are still images with the same backgrounds. But what we really care about are not such frames. Therefore, when we want to search for or protect some special frames in these videos, we should develop more effective methods.

Searching for target persons over a video by humans is time-consuming, because the persons must be found everywhere in the image sequence. Even in a shorter segment of videos to search some frames is not an easy work, either. Therefore, we should add some marks in special frames to help us in the search. One useful approach is to embed special features into corresponding frames.

In parking lots what we are interested in is the occurrence of a special event on a car. Often, this event means that the car door has been opened. Therefore, if the car is able to send some message out, one useful approach is to embed that message into corresponding frames of the video taken by a surveillance camera in the parking lot. This would be of great help for humans to find the event afterward from the surveillance video.

Because a surveillance video might contain suspicious or criminal acts, malicious users might acquire the video in an illegal way and tamper with it for misrepresentation. It is desired to design in this study a useful approach to counteract such actions. And a solution is to embed the special frames including suspicious acts into the other frames which contain just the environment background. Then, if someone only tempers with the important frames with criminal acts but does not do any change to the background frames, which is possibly the case, we can still recover some frames which contain the suspicious acts for use as a proof of the criminal acts found in the video as well a proof of the tampering behavior.

In this study, we will develop appropriate techniques for accomplishing the different purposes mentioned above.

1.2 Survey of Related Works

In this study, we develop some new methods for information hiding applications by embedding a variety of information within videos.

There are lots of approaches to hiding data into a video [1-3]. Chae et al. [1] proposed a method to hide data into the DCT coefficients of a host video, in which the method is adaptive to the local texture content of the host video frame blocks. The method is useful because the human visual system is more sensitive to the change in low frequency regions than in highly textured regions. Hartung et al. [2] proposed methods for embedding additive watermark signals into uncompressed and compressed video sequences. Giannoula et al. [3] proposed a compressive data hiding techniques. This method hides both the chrominance components and the DCT coefficients of certain segmented audio signals into the luminance component.

For video recovery, some authentication techniques have been applied. Chien et al. [4] proposed a method to detect tampering of frames by embedding authentication signals. Schneider et al. [5] proposed an idea to generate a digital signature for authentication using the image content. He et al. [6] proposed an authentication method to authenticate MPEG-4 videos with object-based watermarking solutions.

1.3 Overview of Proposed Methods

1.3.1 A method for searching targeted persons in surveillance videos

A method using the data hiding technique for searching targeted persons in surveillance videos is proposed in this study. First, we apply a motion detection technique to judge whether an image captured from a video camera is a still background frame or not. If not, a color detection algorithm is performed on the image to obtain a certain color feature. Then an MPEG-4 encoder is employed to compress the image, and embed in the mean time the color feature into some quantized DCT coefficients of the I-frame detected to be a non-background frame. Finally, a variable-length coding process is performed to generate a *stego-video* (a video in which some digital information is embedded). By extracting the color feature through an MPEG-4 decoding process, we can easily find out which frame is a non-background one and what color feature it has in the surveillance video. Then if we find a person in one frame of the video, we may find the same person in the other frames of the video by finding frames which have similar color features.

1.3.2 A method for hiding and recovering suspicious images in surveillance videos

A method using data hiding techniques for hiding and recovering suspicious image parts of surveillance videos is proposed in this study. First, we apply a motion detection technique to judge whether an image frame captured from a video camera contains motions or not. If so, we store the motion frame, called *suspicious image*. Then an MPEG-4 encoder is employed to compress the image, and embed the suspicious image into some quantized DCT coefficients of the background I-frames in the following image sequences of the video. After completing the embedding process, a variable-length coding process is performed to regenerate a protected video. By extraction the suspicious image through an MPEG-4 decoding process, we can obtain the suspicious image that we are really concerned with. And if someone has tampered with that part of the image sequence which we have embedded into background frames, we still can get some of those import frames.

1.3.3 A method for searches of special-event frames in surveillance videos of parking lots

A method using data association techniques for searching special events at parking lots by hiding special signal information in surveillance videos is proposed in this study. First, we assume that each car parked in a parking lot is wirelessly connected to the Internet. And when a car door is opened, the car will send an identification signal to a surveillance system established in the parking lot, and the system will hide the signal into the video immediately. In the surveillance system, each image captured from a surveillance camera is put into a real-time encoding process. Car identification signals are embedded into some quantized DCT coefficients of the I-frames in the recorded video when some cars send signals to the system. A variable-length coding process is performed finally to generate a *stego-video*. By extracting the car identification signals through an MPEG decoding process, we can find out which car in the surveillance has been opened and when it was opened.

1.4 Contributions

Several contributions are made in this study, as described in the following.

1. A method for searching targeted persons in recorded videos by hiding and retrieving color feature information is proposed.
2. A method of hiding, and recovering suspicious images in background frames of surveillance videos by motion detection techniques is proposed.
3. A method for searches of special-event frames in surveillance videos of parking lots by hiding special signal information is proposed.

1.5 Thesis Organization

In the remainder of this thesis, the method for searching targeted persons in surveillance videos is described in Chapter 2. In Chapter 3, the proposed method for hiding and recovering suspicious image parts in surveillance videos is described. In Chapter 4, the proposed method for searches of special-event frames in surveillance videos of parking lots is described. Finally, conclusions and some suggestions for future works are given in Chapter 5.

Chapter 2
Searches of Targeted Persons in Surveillance by Hiding Color Feature Information in Surveillance Videos

2.1 Introduction

With rapidly increasing popularity of surveillance systems, more and more crime facts or special events can be recorded and found in surveillance videos. In this chapter, we describe the method we propose for searching surveillance videos for recorded motion events (like a walking person) in video frames. The method is useful because suspected humans and their activities in surveillance videos can then be checked quickly and conveniently for various purposes. Conventionally, when we find some suspicious persons in a video and want to find other video frames which contain the same person, it usually takes a lot of time to do such works manually by humans. Instead, by the method we propose, we can quickly find other video frames which contain similar suspicious persons or activities.

Data hiding techniques have been proposed in many researches recently. Chien and Tsai [4] proposed an approach to embedding data into videos during the encoding process. And in this study we will use their approach in the proposed embedding process.

In Section 2.1.1, some related problem definitions are given, and the idea of the proposed method is described in Section 2.1.2. In Section 2.2, the proposed process for embedding color feature information is stated. In Section 2.3, the proposed process of extracting hidden information and searching of targeted persons is stated. In Section 2.4, some experimental results are shown to prove the feasibility of the proposed method. In the last section, a summary will be made.

2.1.1 Problem Definition

In the proposed method, given a surveillance video, whenever a motion event is found in a video frame, say *I*, we extract a feature *F* of *I* immediately and embed *F* in the first subsequently *I*-frame of the video. In this way, all the motion events can be extracted and saved in the video itself. Later on, whenever search of a certain motion event is desired, we can extract the feature F' of the event from a relevant I-frame, and then use F' as a seed for searching the remaining part of the video to find out all the I-frames containing features similar to F'. This technique is useful for fast search of interesting events in long surveillance videos.

The first issue involved in the proposed method is how to detect motion events in a real-time surveillance system with a stationary camera. The second issue is how to generate a feature to identify frames which contain an identical event. And the third issue is how to embed the feature into compressed bitstreams of MPEG-4, and the final issue is how to extract the feature in the search process.

2.1.2 Proposed Idea

An example of interesting motion events is the appearance of a certain person in many video frames. And usually, these video frames have similar features, like *color* of the clothes of the person. We can use the color feature to detect whether two frames contain the same person. In the proposed system, we utilize *absolute frame differencing*, which compares an input video frame with a fixed background image, to detect motions in the video frame. We then want to embed the color feature of the motion part into the video during a video encoding process. Then, we can easily check which frames contain the same person later on.

2.2 Embedding of Color Feature Information

In the proposed system, we utilize an absolute frame differencing algorithm, which compares an input video frame with a fixed background image, to detect motions in the video frame. In this section, we propose a method to extract a color feature of a non-background video frame, and describe the process for embedding the color feature into the subsequent nearest I-frame in the video. An illustration of the proposed method is shown in Figure 2.1.

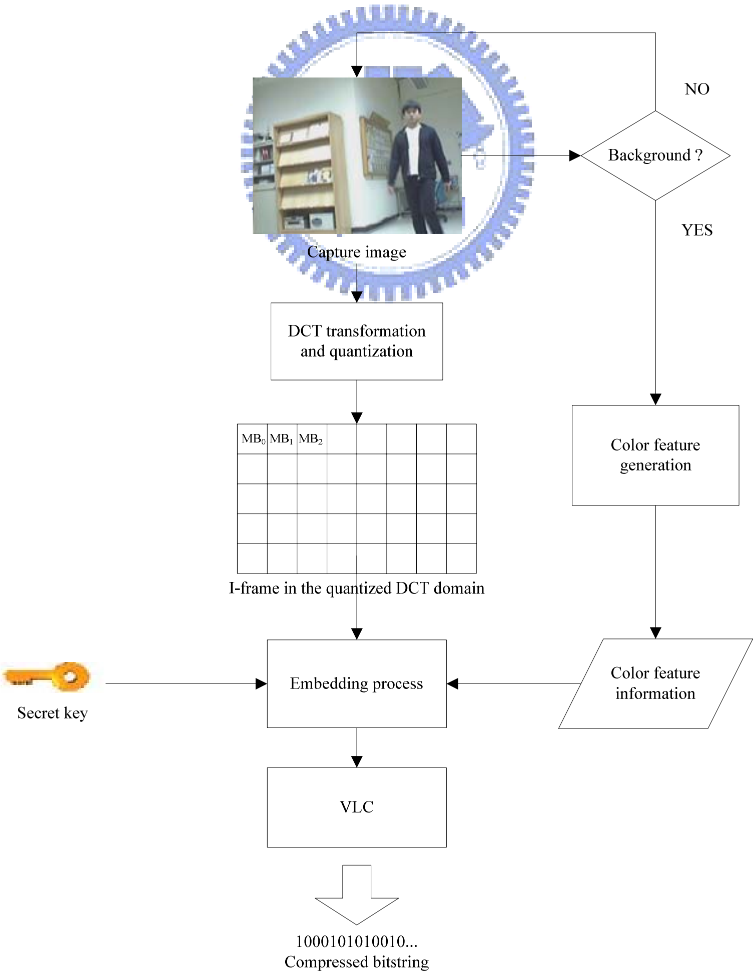


Figure 2.1 Illustration of proposed method for embedding color feature information in real time.

2.2.1 Properties of color feature information

First, we transform the color space into another, the model of “$\mathrm{YC}_b\mathrm{C}_r$.” Then we deal with the chrominance (C_b, C_r) only to generate the desired color feature. In this way we can reduce the effect that the light has on colors. We know the ranges of C_b and C_r are clamped to [1..255]. Now, we divide the space (C_b, C_r) into 256 parts with the same size as Figure 2.1 shows. Then we count how many pixels of an image in each part. And we denote the two parts which have the highest counts as the desired color feature of this image.

(Cb,Cr)	0~15	16~31	…	194~209	210~255
0~15	(0,0)	(0,1)	…	(0,14)	(0,15)
16~31	(1,0)	(1,1)	…	(1,14)	(1,15)
⋮	⋮	⋮	…	⋮	⋮
194~209	(14,0)	(14,1)	…	(14,14)	(14,15)
210~255	(15,0)	(15,1)	…	(15,14)	(15,15)

Figure 2.2 An illustration of divided parts of chrominance space.

2.2.2 Proposed embedding process

In the proposed system, we check each video frame of a captured video to see whether the video frame is a background image or not. If not, we get a difference image by subtracting the video frame from the background image. We do this in the space of (Y, C_b, C_r). Then, we check the three colors Y, C_b, and C_r of each pixel of the difference image; if any of the three values is larger than a corresponding threshold value Ty, Tc_b, or Tc_r, which is predefined, then we alter the pixel values to be those of the video frame image at the corresponding position; otherwise, we alter the three values as null. Finally, the altered difference image will be treated as the input data to Algorithm 2.1. This completes the extraction of the color features of the captured video frame image.

***Algorithm* 2.1**: Process of color feature extraction.

Input: a difference image *D* in the YC_bC_r color space generated from a video frame image *I*.

Output: the color feature *C* of *I*.

Steps:

1. Denote each divided part (*x*, *y*) of the chrominance space of Figure 2.1 as $Z_{x,y}$.
2. Assign a counter to $Z_{x,y}$ and let it be denoted as $C_{x,y}$.
3. Obtain the ranges $R_{b_{xy}}$ and $R_{\mathrm{r}_{xy}}$ of the values of C_b and C_r of $Z_{x,y}$, respectively, i.e., set $R_{b_{xy}}$ as $16x \sim (16x + 15)$ and $R_{\mathrm{r}_{xy}}$ as $16y \sim (16y + 15)$, as can be seen from Figure 2.1.
4. Scan each pixel P_{ij} in *D*, and let its chrominance components be $(C_{b_{ij}}, C_{r_{ij}})$. If the values $C_{b_{ij}}$ and $C_{r_{ij}}$ are in the ranges $R_{b_{xy}}$ and $R_{r_{xy}}$, respectively, then set $C_{x,y} = C_{x,y} + 1$.
5. Select the two largest values of all the counters from $C_{0,0,}$ to $C_{15,15}$, and denote corresponding two divided parts in the chrominance space as Z_{x,y_m} and Z_{x,y_s}.
6. Use the indices (x_m, y_m) and $(x_\mathrm{s}, y_\mathrm{s})$ to compute two numbers $n_m = 16x_m + y_m$ and $n_s = 16x_s + y_s$, transform them into binary numbers, and concatenate the

results in order to form a binary string C as the desired color feature.

After we get the color feature by Algorithm 2.1, we must embed it into the subsequent nearest I-frame in the given video. The detail steps of the embedding process are described in Algorithm 2.2 below. And a corresponding flowchart is shown in Figure 2.2

***Algorithm* 2.2**: The process for embedding the color feature.

Input: an I-frame F in the quantized DCT-domain, a secret key K, and the color feature C of a non-background frame F_m.

Output: a stego-I-frame F' in the quantized DCT-domain.

Steps:

1. Denote the binary form of C as $C_b = b_1b_2b_3\ldots b_L$, where L represents the length of C_b.
2. For each luminance block B of size 8×8 of F, using the input key K to select a pair of DCT coefficients (AC_1, AC_2) from ten pre-selected ones to embed a bit b_k of C_b according to the following rule:

   (1) when $b_k = 0$ and $k \neq L$:

   if $AC_1 < AC_2$, then swap AC_1 and AC_2;

   if $AC_1 = AC_2$, then set $AC_1 = AC_1 + T$;

   (2) when $b_k = 1$ and $k \neq L$:

   if $AC_1 > AC_2$, then swap AC_1 and AC_2;

   if $AC_1 = AC_2$, then set $AC_2 = AC_2 + T$;

   where T is a pre-defined threshold value.

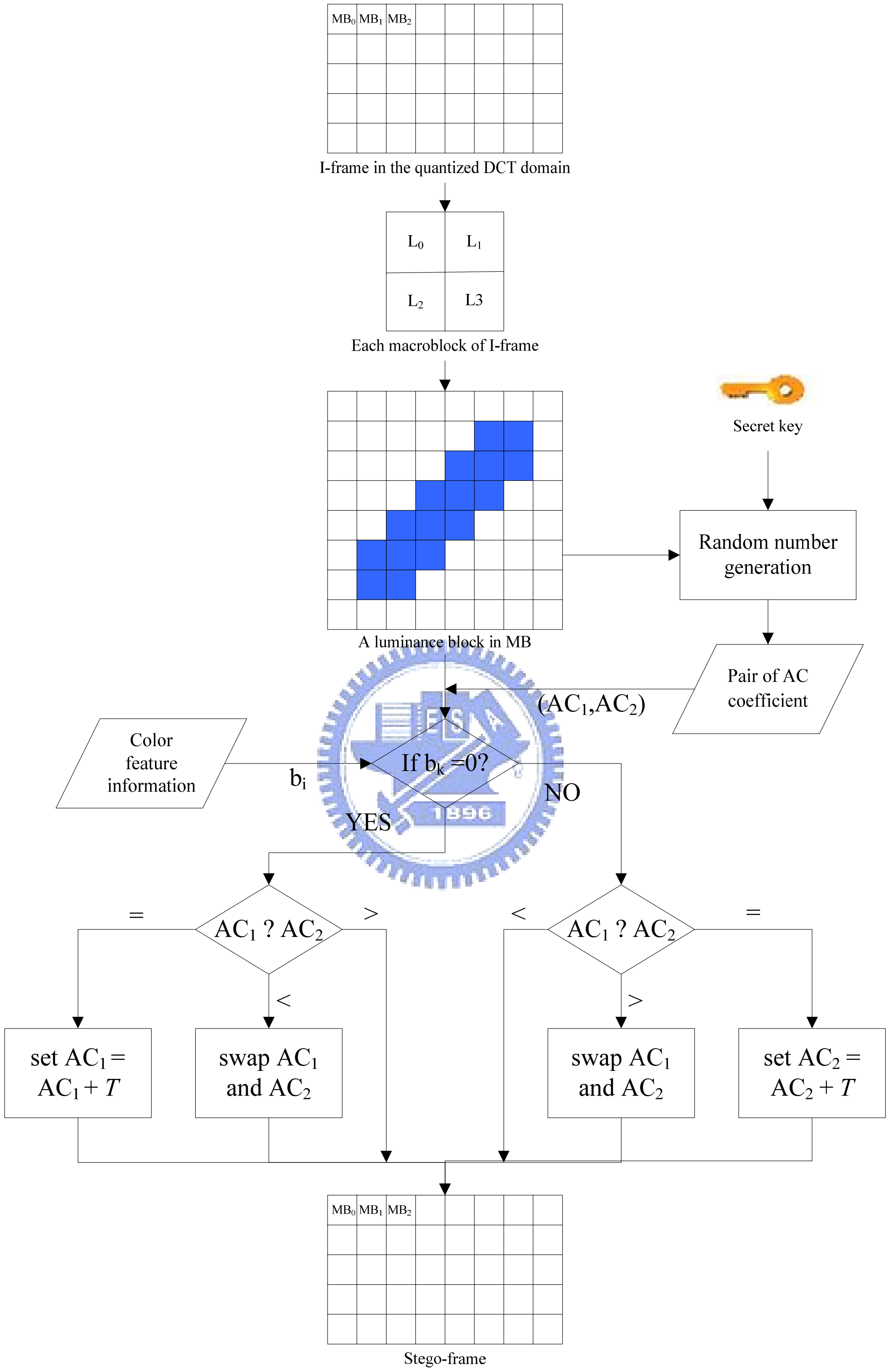


Figure 2.3 Flowchart of process for embedding color feature.

2.3 Extraction of Hidden Information and Searches of Targeted Persons

In Section 2.3.1, the idea behind the proposed color feature extraction process is given. A detailed algorithm of the extraction process is described in Section 2.3.2. And the steps of targeted person searches are described in Section 2.3.3.

2.3.1 Idea of proposed extraction technique

In the last section, we describe how to embed a color feature in an I-frame. Before we use such information for searches of targeted persons, we must extract the color feature from the quantized DCT-domain during the decoding process. For this purpose, we can extract the information only in the compressed domain; we do not have to perform the complete decompression during the extraction process.

2.3.2 Proposed extraction process

In fact, when we do not embed color feature information in an I-frame, we embed instead a character “N” into this I-frame to denote that this frame does not contain any information. In this way, we can check whether a video frame contains a color feature or not. In the following is the algorithm we propose to extract the color feature information from an I-frame in a stego-video. We use this algorithm to extract all the color feature information which has been embedded in the stego-video, and list all of them in a report for search.

***Algorithm* 2.3**: The process for extracting embedded color feature information.

Input: a stego-I-frame F' in the quantized DCT-domain and a secret key *K*.

Output: a color feature *C*′.

Steps:

1. For each 8×8 luminance block B_{ij} of F' with data embedded, use the secret key K to select from the ten pre-defined pairs of DCT coefficients the right pair into which data have been embedded.
2. Denote the selected pair of DCT coefficients as (AC_1, AC_2).
3. Denote the desire color feature information of F' as C'.
4. Extract a bit b_k of C', a binary element of the extracted color feature, according to the following rule:

when $k \le L$:

if $AC_1 > AC_2$, then set $b_k = 0$;

if $AC_1 < AC_2$, then set $b_k = 1$;

where L is the length of C'.

2.3.3 Proposed search process

With Algorithm 2.3 we can get the color features for some frames of a video. If a frame of the video has color feature information, it means that this video frame has information which has been embedded into. In other words, this video frame must have contained some motion image parts. Usually, these motions are caused by human activities. So, it is highly possible that the color feature information is the feature of a person who causes the motion. Therefore, if the same person appears in many frames of the video, we may find similar color features in each of these frame. We use this property to search the targeted person. The detailed steps are described in Algorithm 2.4. We can get the color feature of the targeted person as input C in one slice of the video which contains the person.

***Algorithm* 2.4**: The process for searching a targeted person.

Input: all extracted color features (C_1, C_2, …, C_N) of the video and the color feature C of a targeted person.

Output: all slices of the video which have similar color features to *C*, forming a search list *L*.

Steps:

1. Take one of the color features and denote it as $\mathrm{C}_{k,}$ $k = 1, 2, \ldots, N$.
2. Denote the information of C_k as I_k (including the position of C_k in the video and time, etc.)
3. Obtain the two components of C_k and denote them as M_k and S_k. Also, obtain the two components of C and denote them as M and S.
4. Compare C with C_k to search frames of the video having similar color features according to the following rule:

> if $M_k = M$ or $M_k = S$ or $S_k = M$ or $S_k = S$, then add the video frame I_k to the search list L.

2.4 Experimental Results

In our experiments, we used a web camera and a notebook to simulate a surveillance system. Each image captured by the camera was encoded in real time by an encoder to form a compressed video with frame size 320×240. Figure 2.4 shows the difference Figure 2.4(b) of one captured frame Figure 2.4(a).Figure 2.5 shows six consecutive frames of the resulting stego-video, with the color feature of Figure 2.4(a) of the captured frame being embedded in the I-frame (the 4th frame shown in Figure 2.5(d)). Figure 2.6 shows the result of extracting the color feature and searching for a targeted person by a user interface. We select a color feature from a frame of the resulting stego-video, i.e., Figure 2.6(a), as the seed for the search. Then we tried to find other similar color features in the stego-video. Figure 2.4(b) shows the search result corresponding to the selected color feature of Figure 2.6(a). Totally four frames with similar features were extracted.

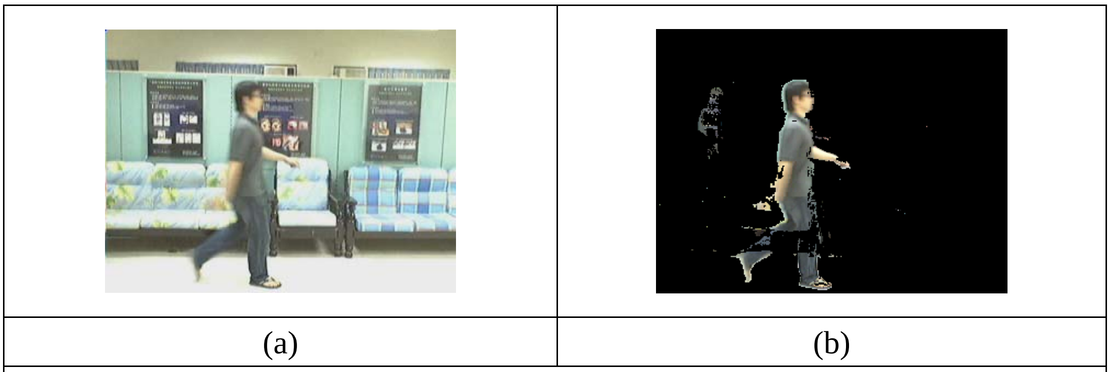

(a) (b)

Figure 2.4 An image and difference with background. (a) Original image. (b) Difference image.

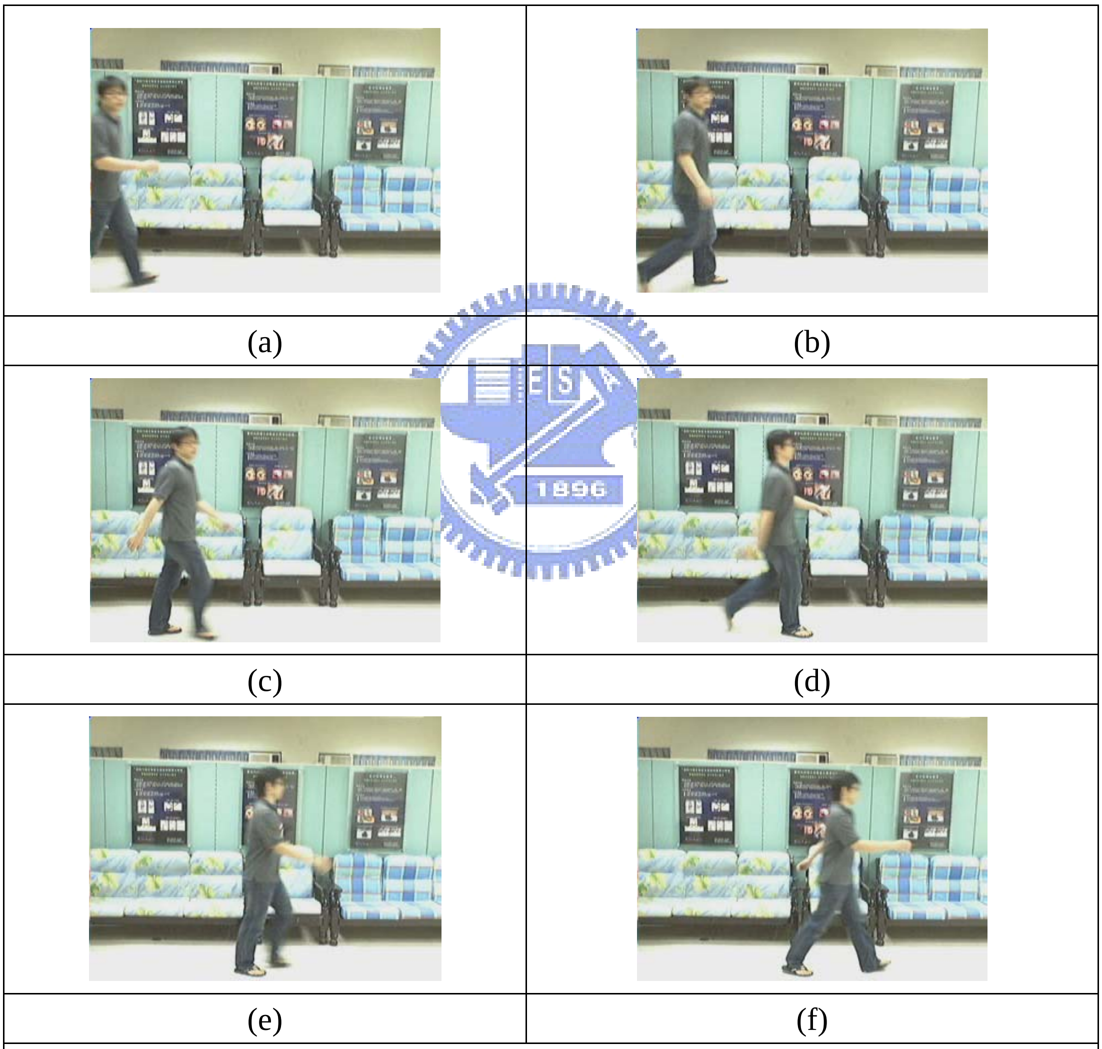
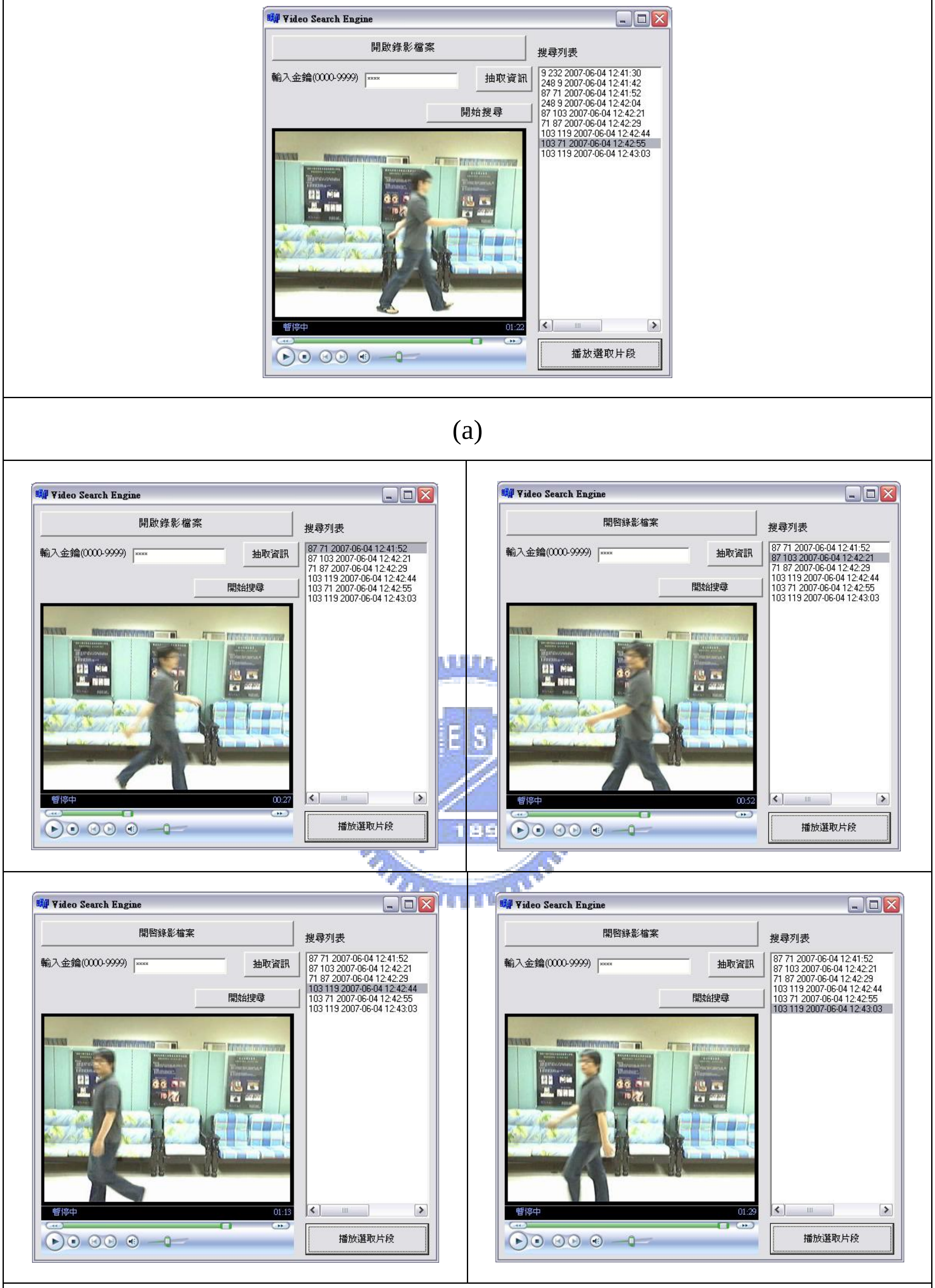

(a) (b) (c) (d) (e) (f)

Figure 2.5 Six frames of the resulting stego-video. (a) The first frame (P frame). (b) The second frame (B frame). (c) The third frame (P frame). (d) The 4th frame (I frame). (e)The 5th frame (B frame). (f)The 6th frame (P frame).(continued)



輸入金鑰(0000-9999) ****

9 232 2007-06-04 12:41:30
248 9 2007-06-04 12:41:42
87 71 2007-06-04 12:41:52
248 9 2007-06-04 12:42:04
87 103 2007-06-04 12:42:21
71 87 2007-06-04 12:42:29
103 119 2007-06-04 12:42:44
103 71 2007-06-04 12:42:55
103 119 2007-06-04 12:43:03

(a)

輸入金鑰(0000-9999) ****

87 71 2007-06-04 12:41:52
87 103 2007-06-04 12:42:21
71 87 2007-06-04 12:42:29
103 119 2007-06-04 12:42:44
103 71 2007-06-04 12:42:55
103 119 2007-06-04 12:43:03

輸入金鑰(0000-9999) ****

87 71 2007-06-04 12:41:52
87 103 2007-06-04 12:42:21
71 87 2007-06-04 12:42:29
103 119 2007-06-04 12:42:44
103 71 2007-06-04 12:42:55
103 119 2007-06-04 12:43:03

輸入金鑰(0000-9999) ****

87 71 2007-06-04 12:41:52
87 103 2007-06-04 12:42:21
71 87 2007-06-04 12:42:29
103 119 2007-06-04 12:42:44
103 71 2007-06-04 12:42:55
103 119 2007-06-04 12:43:03

輸入金鑰(0000-9999) ****

87 71 2007-06-04 12:41:52
87 103 2007-06-04 12:42:21
71 87 2007-06-04 12:42:29
103 119 2007-06-04 12:42:44
103 71 2007-06-04 12:42:55
103 119 2007-06-04 12:43:03

(b)

Figure 2.6 The proposed interface displays the experimental result (a) The result of extraction. (b) The results of searching a targeted person.

2.5 Discussions and Summary

In this chapter, we have proposed a method for a surveillance system to search a video for a targeted person which appears in a preceding frame in the video. In proposed method, we generate a color feature for each motion, and embed it in to the nearest subsequent I-frame. Later on, we extract a certain color feature of interest from the stego-video, which is embedded in an I-frame, and use the feature for searching the video for I-frames with similar color features. In this way, we can quickly know when the person, who is represented by the feature of interest, has appeared so that we can get the corresponding video frames easily. This technique is useful for fast search of videos for targeted persons of interest, saving time in observing a long video for the purpose of finding out all the scene frames including the person.

Chapter 3
Recovery of Modified Surveillance Videos by Hiding Motion Information in Background Frames

3.1 Introduction

With the rapid development of the environment surveillance system, large volumes of digital surveillance videos are produced every day. It is easy to modify digital videos by lots of video editing software. So, criminal or special events recorded in the video may be tampered with by illicit people. In this chapter, we describe the method we propose for video authentication and recovery to prevent such illegal activities.

3.1.1 Problem definition

The first task of the proposed method is to verify whether a given video has been tampered with or not, and identify which frames have be modified or replaced. The second task of the propose system is to reconstruct such frames. The ideas behind the techniques we propose to achieve these two tasks are described in the following.

3.1.2 Proposed idea

In most surveillance videos, there are lots of background frames with no motion activity. These frames usually are of low values to users. But it spends a lot of spaces to record them. In the proposed method we use these spaces to embed more valued information. More specifically, we compress video frames with motion activities (called motion frames hereafter) and embed them together with certain authentication signals in subsequent stationary background frames (called *stationary frames* henceforth). The authentication signals are computed from the contents of the motion frames, and can be used later for checking whether the motion frames have been tampered with or not. Furthermore, if the motion frames are authenticated to have been tampered with, then the original motion frames which were embedded in subsequent stationary background frames can be retrieved for inspection for various purposes, for example, as proof of the illegal tampering activity.

3.2 Embedding of Motion Frames and Authentication signals into Stationary Frames

In the proposed system, we utilize the technique of *relative frame differencing*, which compares an input video frame image with its preceding frame image, to detect motions in the input frame image. And we compress the detected motion frames and embed the resulting data as well as the authentication signals into subsequent stationary frames. More details are described in the following.

3.2.1 Properties of motion frame images and proposed authentication signals

In the proposed video surveillance system, consecutive frames are first encoded into the MPEG format in the form of *GOPs* (groups of pictures). Each GOP includes an I-frame, and several B-frames or P-frames, in order, with each frame being in the form of a bit stream. Before the encoding process, if a frame acquired by a surveillance camera is analyzed by the previously-mentioned relative frame differencing technique to be a motion frame, then the encoded GOP which includes this frame will be regarded to be a *motion GOP*. To enhance security, we use a secret key in the process of authentication signal generation and motion frame embedding. In this way, other people cannot generate the same signals and retrieve the embedded data without the right key.

3.2.2 Proposed authentication signal generation process

We describe the process of authentication signal generation as an algorithm in the following.

***Algorithm* 3.1**: Generation of an authentication signal.

Input: a GOP G_i of the captured video, the frame number N of the I-frame of the GOP G_{i+1} following G_i, and a secret key K.

Output: authentication signal A_i of G_i.

Steps:

1. Denote the n frames of G_i as F_{i0}, F_{i1}, F_{i2}, …, F_{in}, where F_{i0} is an I-frame, and n is the number of frames in G_i.
2. Encode these n frames into the MPEG format, and denote the results as B_{i0}, B_{i1}, …, B_{in}.
3. Concatenate B_{i0}, B_{i1}, …, B_{in}, and K in order as a string B_i, and calculate the CRC value of B_i as $GCRC_i$.
4. Concatenate B_{i0} and K as I_i, and calculate the CRC value of I_i as $ICRC_i$.
5. Concatenate $ICRC_i$, $GCRC_i$, and N in the form of a bit stream as the desired output A_i.

3.2.3 Proposed embedding process

In the proposed system, we consider a GOP constructed from motion frames as a basic unit and we embed all of the data of each GOP only in the subsequent I-frames. If one frame of a GOP is a motion frame, we set this GOP as a motion GOP. The following is the proposed algorithm for constructing the data to be embedded into an I-frame.

***Algorithm* 3.2**: Construction of data to be embedded in an I-frame.

Input: a GOP G_i and the nearest subsequent I-frame I_{i+1}.

Output: the data D_i which will be embedded in I_{i+1}.

Steps:

1. Generate the authentication signal A_i of G_i by Algorithm 3.1.
2. Decide whether G_i is a motion GOP or not by checking if there is at least one frame in G_i which is a motion frame.
3. Define a flag F whose value is determined in the following way:

   if G_i is a motion GOP or I_{i+1} is a motion frame, then set F = “M” (meaning that only the authentication signal A_i is to be embedded in I_{i+1});

   otherwise, set F = “N” (meaning that both the motion GOP and the authentication signal A_i are to be embedded in I_{i+1}).
4. Decide the *video-recovery information* R_i to be embedded in I_{i+1} by the following rule:

   if F = “M,” then set R_i = ‘null’ (meaning nothing to be embedded);

   if F = “N,” then set $R_i = PG_k$, which is a *fixed part* of the bit stream of G_k, where $k < i$ and G_k is one of the motion *GOP* which has been encoded before and saved in a queue at the end of the queue (if the queue is empty, then set set R_i = ‘null.’

5. Obtain the length of PG_k, the position of PG_k in G_k, and the length of G_k, and concatenate all of them as IN_i (meaning the information of R_i).
6. Concatenate A_i, R_i, and IN_i as D_i.

The algorithm of embedding data in the I-frame is the similar as Algorithm 3.2, but we use all of the ten pre-selected pairs of DCT coefficients. Since we can protect the authentication signal by a key, we do not need to use the key to select these pairs.

***Algorithm* 3.3**: Embedding of the authentication signal and the video-recovery information.

Input: an I-frame F in the quantized DCT-domain, and data D generated by Algorithm 3.2 to be embedded into F.

Output: a stego-I-frame F' in the quantized DCT-domain.

Steps:

1. Convert D into a binary form $D_b = b_1b_2b_3\ldots b_L$, where L represents the length of D_b.
2. For each luminance block B of size 8×8 of F, select the ten pre-selected pairs of DCT coefficients, with each pair, (AC_1, AC_2), to embed a bit b_k of D_b according to the following rule:

   (3) when $b_k = 0$:

   if $\mathrm{AC}_1 < \mathrm{AC}_2$, then swap AC_1 and AC_2;

   if $\mathrm{AC}_1 = \mathrm{AC}_2$, then set $\mathrm{AC}_1 = \mathrm{AC}_1 + T$;

   (4) when $b_k = 1$:

   if $\mathrm{AC}_1 > \mathrm{AC}_2$, then swap AC_1 and AC_2;

   if $\mathrm{AC}_1 = \mathrm{AC}_2$, then set $\mathrm{AC}_2 = \mathrm{AC}_2 + T$,

   where T is a pre-defined threshold value.

3. If the all the bits in D_b have been embedded, then stop; otherwise, repeat the above step.

3.3 Extraction and Recovery of Hidden Information

3.3.1 Idea of proposed method

In the proposed system, we can use the authentication signal which we embedded in the I-frame to check whether a *GOP* or an I-frame is the same as the original frames. If not, the original bit streams which were embedded in the subsequent frame may be recovered by extracting their bit streams for various purposes.

3.3.2 Proposed extraction process

In the following is the algorithm we propose to extract embedded data from a stego-video and save them for authentication and other purposes.

***Algorithm* 3.4**: Extraction of embedded data.

Input: a stego-I-frame F' in the quantized DCT-domain.

Output: data D' embedded in F'.

Steps:

5. For each 8×8 luminance block B_{ij} of F' with data embedded, extract ten bits from the ten pre-selected pairs of DCT coefficients by the following steps.

   (1) Denote each of the selected pairs of DCT coefficients as (AC_1, AC_2).

   (2) Extract a bit b_k of D' according to the following rule:

when $k \leq L$:

if $\mathrm{AC}_1 > \mathrm{AC}_2$, then set $b_k = 0$;

if $\mathrm{AC}_1 < \mathrm{AC}_2$, then set $b_k = 1$;

where L is the length of D'.

3.3.3 Proposed authentication process

After extracting data from a stego-video, we can get all of the authentication signals from the video. We recalculate the *CRC* value of each GOP in a given suspicious stego-video and that of each I-frame with the given key, and compare the result with the associated authentication signal. If they match, we compare the I-frame number with the extracted associated frame number. If the two comparisons are both good, we can assure that the GOP is correct. If comparisons fail, we decide that it has been tampered with. The following is the process to authenticate a given video with the extracted data.

***Algorithm* 3.5**: Authentication of a given stego-video.

Input: a stego-video V and the data D extracted from V using Algorithm 3.4.

Output: a report R^s including staring I-frame numbers whose corresponding GOPs are not tampered with.

Steps:

1. Obtain the authentication signal A_i of each *GOP* G_i from D.
2. Recalculate the authentication signal of each *GOP* G_i^s of V, and denote the result as C_i^s.
3. For each GOP G_i^s of V, compare A_i and C_i^s to detect whether G_i^s has been tampered with or not by the following rule:

   if the CRC values in A_i and C_i^s are identical, then extract the I-frame number from C_i^s and add it to the report R^s;

else, do nothing.

3.3.4 Proposed recovery process

When we fail to authenticate a *GOP* G_i^s as described previously (i.e., the I-frame number of G_i^s is not in the report R^s yielded by Algorithm 3.5), we check further, using the GOP number, the extracted data D mentioned in the input of Algorithm 3.5 to see whether D contains G_i^s or not. If so, we proceed to recover the authentication-failing *GOP* G_i^s. If the part of the video where G_i^s is located is not tampered with, then we can extract the GOP there as the recovery-information. In this way, the missing GOP can be recovered for inspection.

3.4 Experimental Results

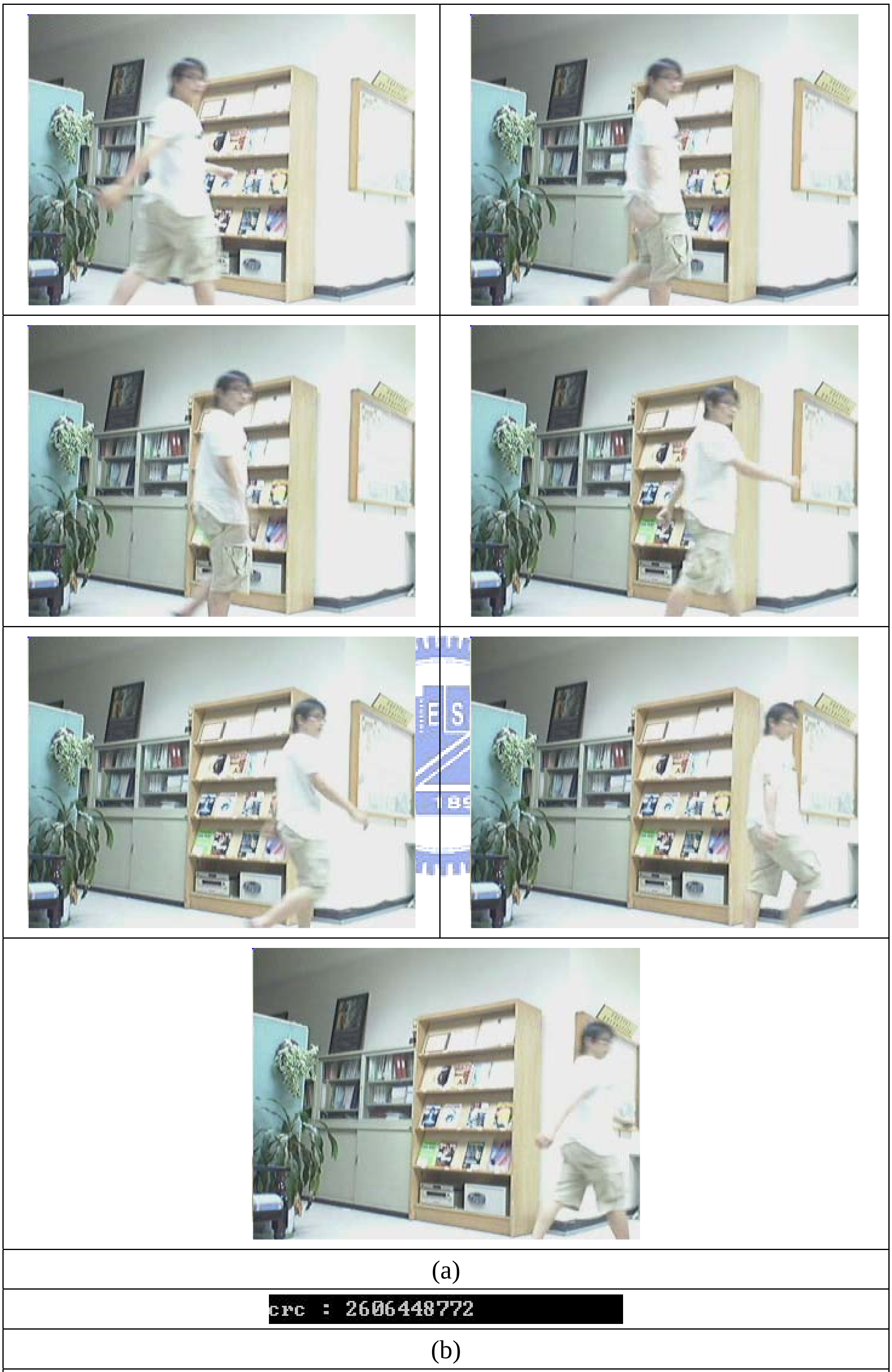
In our experiment, we capture image frames with a web camera and encode the frames in real time into MPEG files. When a captured frame F was checked to include motion parts, the system will store the GOP which includes F in a queue, and embed all the motion GOPs in subsequent background frames. Figures 3.1 through Figure 3.3 shows some frames of our experimental results.

In Figure 3.1(a), we show a motion GOP which consists of 7 motion frames. Figure 3.2(b) shows the computed CRC value of the GOP. In Figure 3.2(a), we show some original background I-frames in which G_1 is to be embedded, and Figure 3.2(b) shows the stego-I-frames in which G_1 is already embedded. It can be observed that there is no noticeable difference created in the stego-I-frames.

Figure 3.3 shows some results involved in a video tampering process and an authentication process. The 7 motion frames of the GOP G_1 were replaced by 7 stationary frames of a new GOP G_1’ in a simulation of tampering with G_1 in our experiment. Figure 3.3(a) shows the 7 stationary frames of G_1’. Figure 3.3(b) shows the computed CRC value of the new GOP G_1’, which is different from the extracted CRC value of GOP G_1 extracted from the subsequent I-frame and shown in Figure 3.3(c). This difference in the two CRC values makes the authentication of the GOP G_1 fail, leading to the action of retrieving the original GOP G_1 which is stored in subsequent I-frames. The retrieved 7 frames of the original GOP G_1 are shown in Figure 3.3(d). This demonstrates the entire processes of authentication and data recovery.

3.5 Summary

In this chapter we have proposed a method of video authentication and recovery. In the proposed system, we use a data hiding technique to embed authentication signals and frame recovery information into a video. In this method, we use the background frames for embedding recovery information. By extracting authentication signals, we can verify whether a given video has been tampered with or not. If so, we can reconstruct the video by the extracted recovery information, which was embedded beforehand.

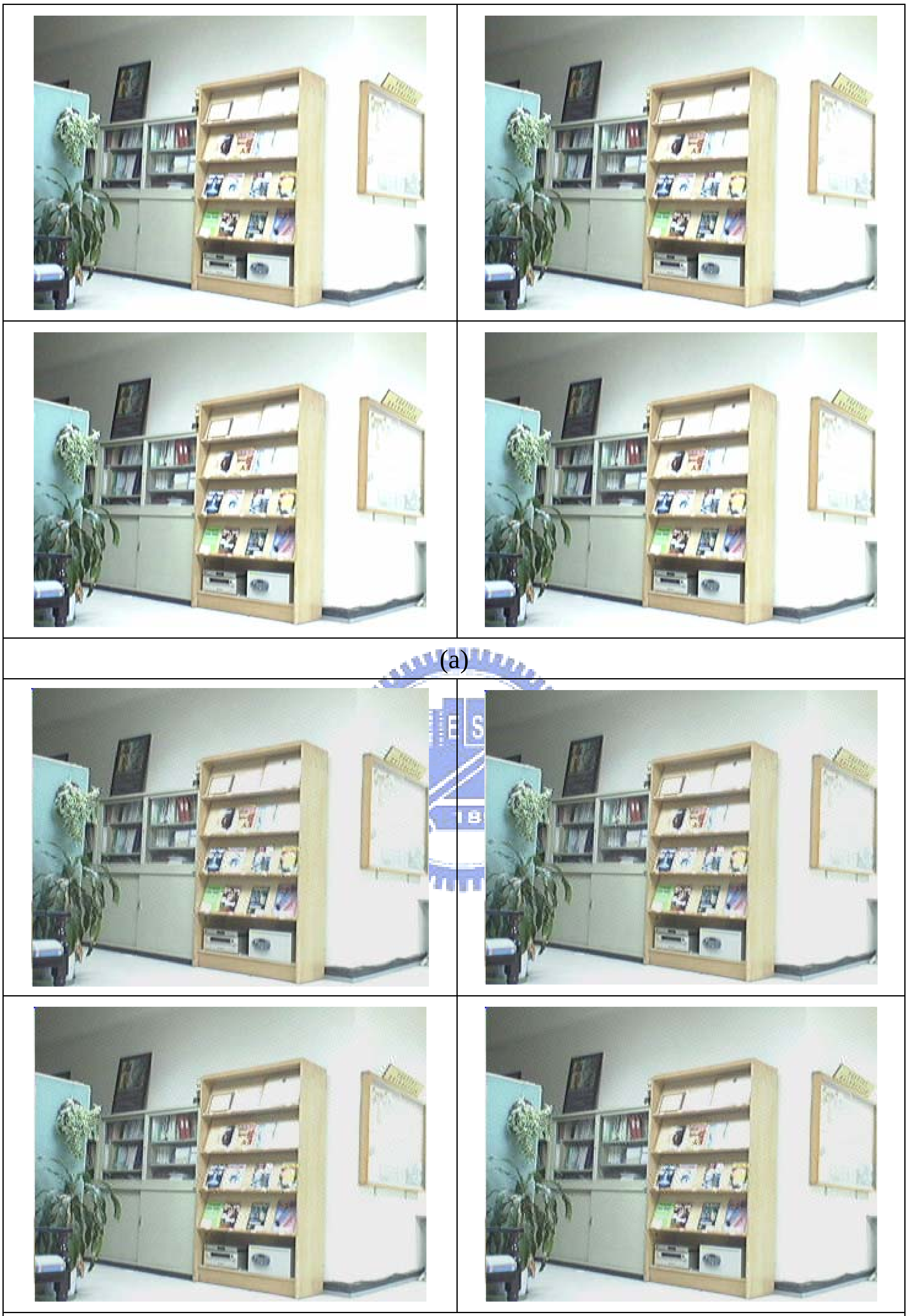


(a)

```
crc : 2606448772
```

(b)

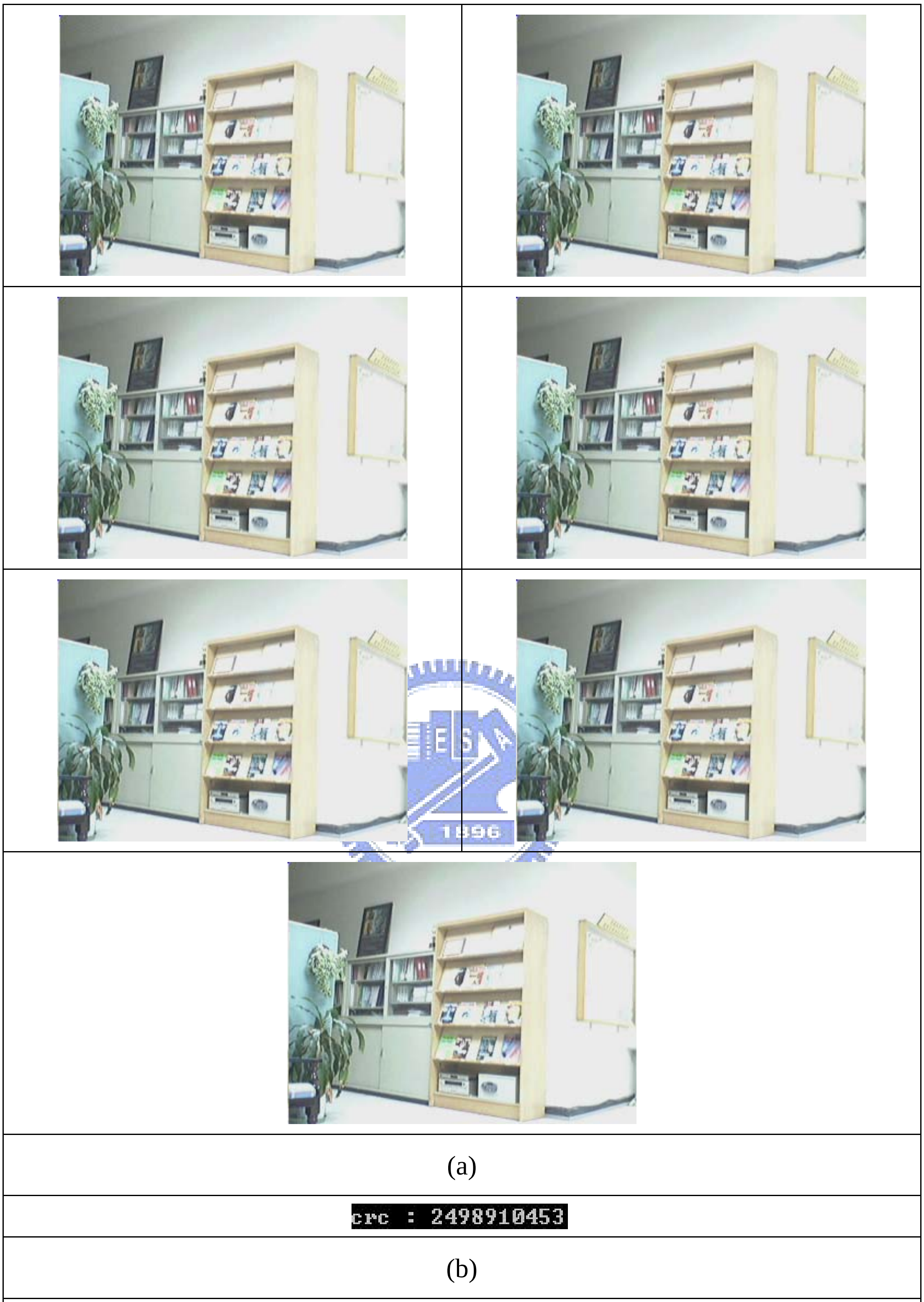
Figure 3.1 A GOP which is embedded in background frames. (a) GOP G_1 with 7 frames. (b) CRC value of G_1 which is a 32-digit integer.



(a)

(b)

Figure 3.2 Several background I-frames in which compressed GOP G_1 of Figure 3.1 is embedded. (a) Original background I-frames in which G_1 is to be embedded. (b) Stego-I-frames in which G_1 is already embedded.

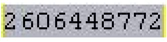


(a)

```
crc : 2498910453
```

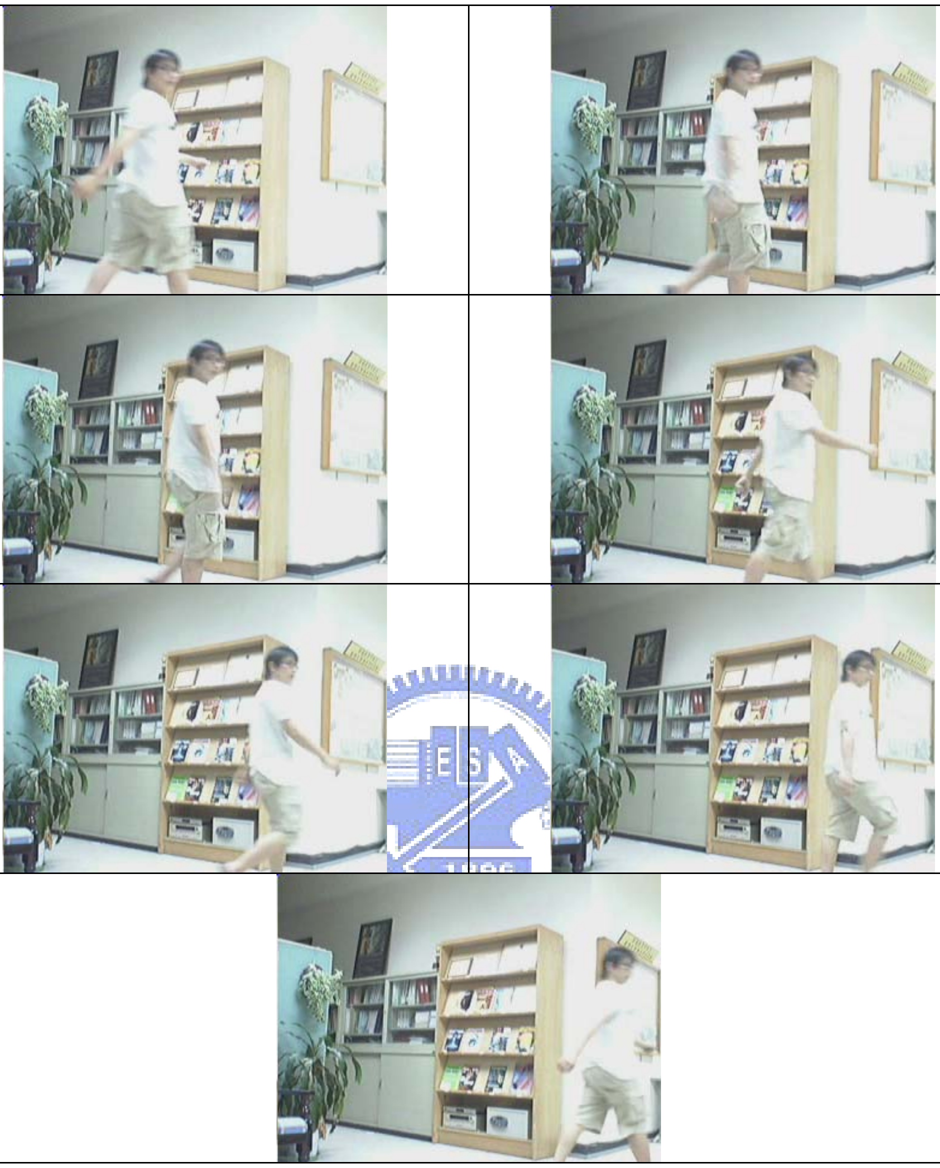
(b)

Figure 3.3 Authentication signal extracted. (a) 7 stationary frames forming a new GOP G1’ which replaces the original 7 frames of GOP G1 (b) Computed CRC value of GOP G1’. (c) CRC value of GOP G1 extracted from the subsequent I-frame. (d) 7 frames of the original GOP G1 extracted from the I-frames where they were embedded.



2606448772

(c)



(d)

Figure 3.3 Authentication signal extracted. (a) 7 stationary frames forming a new GOP G_1' which replaces the original 8 frames of GOP G_1 (b) Computed CRC value of GOP G_1'. (c) CRC value of GOP G_1 extracted from the subsequent I-frame. (d) 7 frames of the original GOP G_1 extracted from the I-frames where they were embedded.(continued)

Chapter 4
Searches of Special Event Frames in Surveillance Videos of Parking Lots by Hiding Car Identification Information

4.1 Introduction

Nowadays, cars are popular everywhere. And parks lots are also around our living environments with many video surveillance systems to protect our cars. When a car has been broken in a parking lot or something in the car has been stolen, the owner of the car and the police usually will ask for the recoded video in the surrounding surveillance system to search the video sequentially for frames which contain the car break-in event. This traditional action of video search frame by frame to find out ‘suspected frames’ is time consuming. In this chapter we propose a system for fast search of special events in surveillance videos.

4.1.1 Problem definition

In this study, we want to speed up the search of the frames from a recoded video, which contain special events of car break-in. The first task to accomplish this goal is to let the surveillance system know when such special events happen. The second is to record the events in the video. The final task is to use the recorded information to find out ‘suspected frames.’

4.1.2 Proposed idea

With the rapid development of the Internet and related network techniques, it is possible to put a computer system on a car and have it connected wirelessly to the

Internet or a local network. Then, when a break-in event happens to the car, the car computer can send a message to a video surveillance system nearby. And the surveillance system can then record the time and the car identification information and use the information to 'label' the 'suspected frames' including the scene of the event. Since each car has a plate number, we can use it as the car identification information. Also, we accomplish the 'labeling' work for the suspected frames by information hiding techniques in this study. That is, we embed the previously-mentioned information of the event time and car plate number in real time into the recorded suspected frames of the surveillance video. Then later on, we can search the video and retrieve the event frames by extracting just the 'labeled' video frames using the embedded information. In this way the search is speeded up since unlabeled frames can be ignored in the search process.

4.2 Proposed Method

In this section, we describe the details of the proposed method described briefly in the previous section. Briefly speaking, we embed the event time and car identification information data into the I-frames of the video during a real-time encoding process which transforms the sequential raw images (BMPs) taken by a web camera into an MPEG video file. The schema of transmitting the event time and car identification information to the surveillance system sever from the car is described in Section 4.2.1. Section 4.2.2 describes the process of embedding the event time and car identification information into a video as 'labels'. In Section 4.2.3, the process of extracting embedded data from a stego-video is described. Finally, the search of suspected frames in a stego-video is described.

4.2.1 Transmitting information to surveillance system sever from a car

We assume that each car in a parking lot has a computer with wireless communication ability to connect to a surveillance system nearby, and that the car computer ‘knows’ the IP address of the surveillance system server. And when some break-in event happens, like the case that the car door has been opened, we enable the car computer to have the ability to send a message (the event time and car plate number) to the server. And the server will embed this message in real time into the surveillance video during the image encoding process. In our experiment, we use a notebook PC to simulate the car computer and press a keyboard to represent the occurrence of an event as well as the triggering of the transmission of the relevant message. That is, after the key is pressed, the message will be sent to the surveillance system sever which we use another notebook PC to simulate.

4.3 Embedding of car identification information into a video

In the proposed embedding process, we combine the event occurrence time and the car plate number to form a binary string N to represent an *event message*. The algorithm for embedding N is similar to Algorithm 2.2. We use a secret key to select one of ten pre-selected coefficient pairs in the quantized DCT domain of the MPEG video to embed a bit of N into the nearest subsequent I-frame in real time. Algorithm 3.3 describes the detailed process. We apply the algorithm to each I-frame when it is being encoded. If no message is sent at the moment the frame is encoded, we embed a flag “N” to indicate that no data are embedded in this I-frame.

***Algorithm* 4.1**: embedding of event message data.

Input: an I-frame F in the quantized DCT-domain, an event message M sent from a car, and a secret key K.

Output: a stego-I-frame F' in the quantized DCT-domain.

Steps:

4. Represnt M in binary form as $M = b_1b_2b_3\ldots b_L$, where L represents the length of M.

5. For each luminance block B of size 8×8 of F, use the input key K to select randomly a pair of DCT coefficients (AC_1, AC_2) from the ten pre-selected ones.

6. Embedding a bit b_k of M into (AC_1, AC_2) according to the following rule:

   (5) when $b_k = 0$ and $k \neq L$:

   if $\mathrm{AC}_1 < \mathrm{AC}_2$, then swap AC_1 and AC_2;

   if $\mathrm{AC}_1 = \mathrm{AC}_2$, then set $\mathrm{AC}_1 = \mathrm{AC}_1 + T$;

   (6) when $b_k = 1$ and $k \neq L$:

   if $\mathrm{AC}_1 > \mathrm{AC}_2$, then swap AC_1 and AC_2;

   if $\mathrm{AC}_1 = \mathrm{AC}_2$, then set $\mathrm{AC}_2 = \mathrm{AC}_2 + T$,

   where T is a pre-defined threshold value.

4.3.1 Extraction of hidden information

In order to use the embedded data for later search of event frames, we should extract the data from the stego-video. We use the following algorithm to check whether an I-frame contains embedded data or not, and extract all of the embedded data from a stego-video.

***Algorithm* 4.2**: extraction of embedded data in a stego-video.

Input: a stego-video V and a secret key K.

Output: report R of extracted data.

Steps:

6. For each I-frame I_i in V, transform it into the quantized DCT-domain as D_i.

7. Denote the data to be extracted from I_i as M_i.

8. For each 8×8 luminance block B_{ij} of D_i with data embedded, using the input key K to select randomly a pair of DCT coefficients (AC_1, AC_2) from the ten pre-selected ones.

9. Extract a bit b_k of M_i, a binary element of the extracted data, according to the following rule:

   when $k \leq L$:

   if $\mathrm{AC}_1 > \mathrm{AC}_2$, then set $b_k = 0$;

   if $\mathrm{AC}_1 < \mathrm{AC}_2$, then set $b_k = 1$.

   where L is the length of M_i.

10. Check the flag F of M_i to decide if M_i contains an event message and report the extracted data by the following rule:

    if F = “N,” then do nothing;

    otherwise, divide M_i into the event time T_i and the car plate number N_i and form a pair (T_i, N_i), and put it into report R.

4.3.2 Search of Suspected Frames

After extracting the data and form a report R, we can use this information to search for the event frames we want. After keying in the car plate number in to a search interface designed in this study, we can quick search the suspected frames related to the car by comparing the car plate number with the data in R. The following is the algorithm devised for this purpose.

***Algorithm* 4.3**: Searching for suspected frames related to a car with an input car plate number.

Input: extracted report R and car plate number N.

Output: suspected frames associated with N put in a search result list L.

Steps:

1. For each extracted pair $P_i = (T_i, N_i)$ which is stored in R, extract the corresponding frame I_i.
2. Set up the search list L by checking each pair $P_i = (T_i, N_i)$ in R according to the following rule:

   if $N_i = N$, then add I_i to L;
   else, do nothing.

4.4 Experimental Results

In our experiment, we use two notebook PCs to simulate two cars parked in a parking lot and a third notebook PC to simulate the server of a surveillance system, as shown in Figure 4.1 (the server is not seen in the figure). We press the keyboard of either of the two notebook PCs in the figure to represent a car break-in event, like car door being opened. The keyboard pressing will cause the notebook PC to send an event message, which includes the event time and the car plate number (i.e., XX-123 or XX-456 in Figure 4.1), wirelessly to the server of the surveillance system, on which a user interface is designed for searching a stego-video for event frames of the concerned car.

To search any event frame of a certain car in a given stego-video, we type the car plate number as input into the user interface, and the server system we design will search the entire stego-video by the previously-described algorithms and display the occurrence times of all relevant event frames related to the specified car on the interface, as shown in Figure 4.2. Then, the user can check the event frames one by one by specifying an event time on the interface, as shown by Figure 4.2(a) in which video frames of two events related to the car with plate number XXX-123 are displayed, or as shown by Figure 4.2(b) in which similar frames related to the car with plate number XXX-456 are displayed.

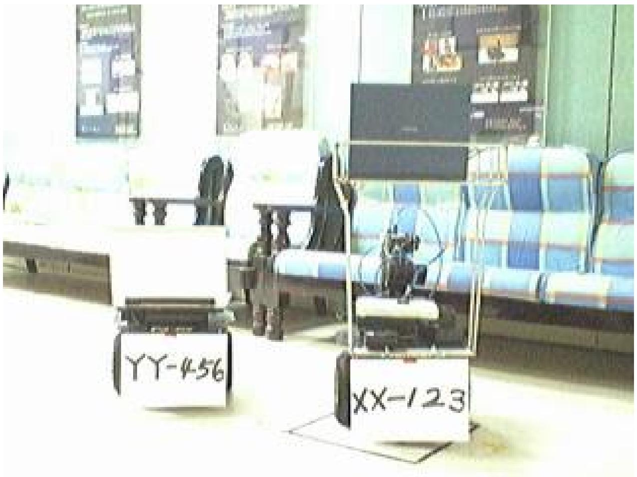
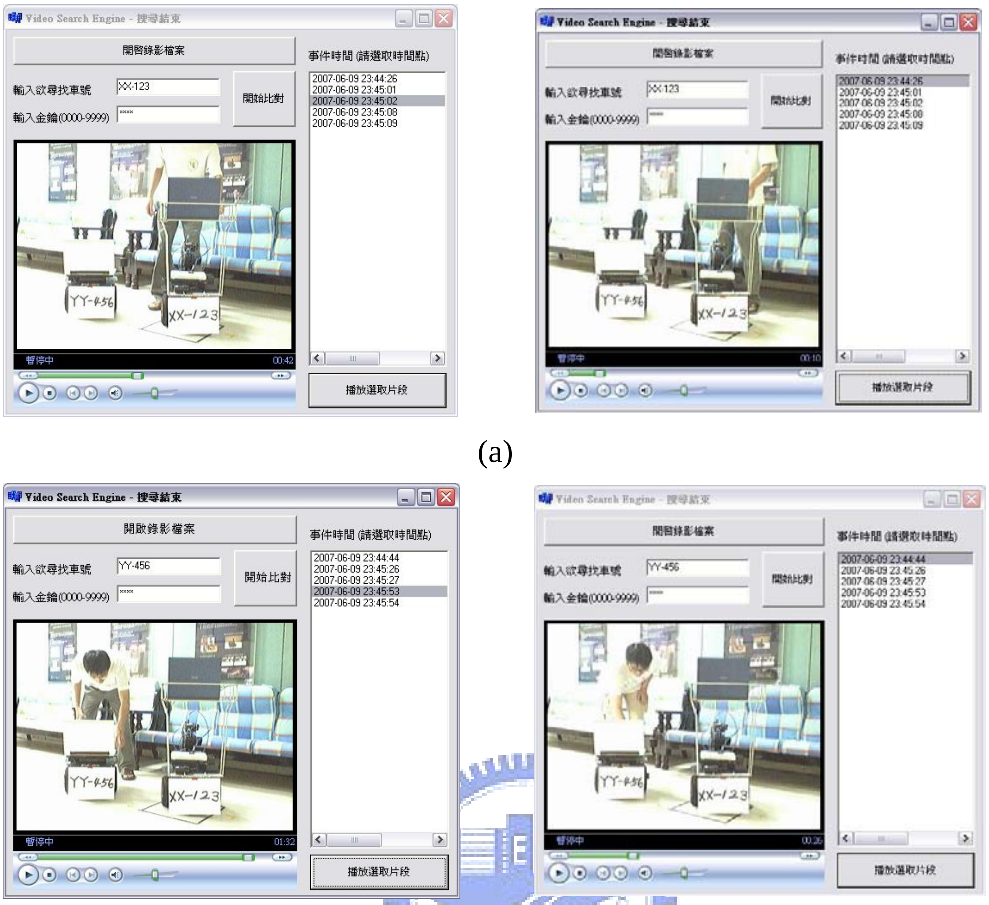


Figure 4.1 Simulating two cars parked in a parking lot.

4.5 Discussions and summary

In this chapter we propose a method to search a stego-video for frames with car events like car stealing or break-in in parking lots. We assume that the car in parking lots can be connected wirelessly to a video surveillance system server through the Internet or a local computer network. And the car can send a message to the server of the surveillance system. The message includes the car identification number (usually the car plate number) and the occurrence time of the event. Then the surveillance system embeds the message into the surveillance video in real-time. Finally we use the data extracted from the stego-video to help searching the video for the frames including the event.



輸入欲尋找車號 XX-123

輸入金鑰(0000-9999) ****

事件時間 (請選取時間點)

2007-06-09 23:44:26
2007-06-09 23:45:01
2007-06-09 23:45:02
2007-06-09 23:45:08
2007-06-09 23:45:09

輸入欲尋找車號 XX-123

輸入金鑰(0000-9999) ****

事件時間 (請選取時間點)

2007-06-09 23:44:26
2007-06-09 23:45:01
2007-06-09 23:45:02
2007-06-09 23:45:08
2007-06-09 23:45:09

(a)

輸入欲尋找車號 YY-456

輸入金鑰(0000-9999) ****

事件時間 (請選取時間點)

2007-06-09 23:44:44
2007-06-09 23:45:26
2007-06-09 23:45:27
2007-06-09 23:45:53
2007-06-09 23:45:54

輸入欲尋找車號 YY-456

輸入金鑰(0000-9999) ****

事件時間 (請選取時間點)

2007-06-09 23:44:44
2007-06-09 23:45:26
2007-06-09 23:45:27
2007-06-09 23:45:53
2007-06-09 23:45:54

(b)

Figure 4.2 The search result with an interface. (a) Display of two of search result of frames of events related to car with plate number XX-123. (b) Display of two of search result of frames of events related to car with plate number XX-456.

Chapter 5
Conclusions and Suggestions for Future Works

5.1 Conclusions

In this study, we have proposed three methods of information hiding in surveillance videos.

In the first method, we search a video for motion frames with color features similar to that of a given frame. By using data hiding technique, such features, which are extracted from the motion parts of motion frames, are embedded into the frames themselves in real time. Since the same person usually wears the same clothes in a day and may appear in a video several times, searching of this way can help us to quickly find out the frames which contain the persons.

In the second method which is useful for recovery of motion frames in a surveillance video, we propose the idea of embedding more valued motion frames into stationary background frames. And we also have proposed an authentication method to verify whether a video has been tampered with. If so, the authentication process will figure out which frames have been attacked. Since we know which frames are incorrect, we may recover the missing motion frames by extraction of them in background frames.

The third proposed method records information about car break-in events in parking lots into surveillance videos for later retrieval. The event message information includes the event time and the car plate number. When a break-in occurs and causes car door opening, the computer on the car will send the message wirelessly to the server of the video surveillance system nearby in the parking lot. Then the server will embed the message as an index in real time into the nearest I-frame which contains the event scene. Later on, if a user wants to search the video for events related to a certain car, he/she may input the car plate number into a user interface of the server to search for the frames involving the specified car. The frames can then all be retrieved for inspection.

5.2 Suggestions for Future Works

Several suggestions for future research works are listed as follows.

1. The proposed video recovery method may use hiding methods with higher data embedding capability to embed more copies of motion frame data. In this way, the probability of all the data copies being destroyed will be reduced.
2. The proposed search methods for surveillance videos in this study may be integrated with the proposed video authentication and recovery method to yield a more powerful method for more complicated applications.
3. The method proposed for car monitoring and suspected frame search may be extended for other purposes, such as monitoring the entrances of buildings or other security areas, as long as event occurrence signals may be generated momentarily, sent to surveillance systems immediately, and encoded in real time as indices for later retrieval and inspection.

References

[1] J. J. Chae and B. S. Manjunath, “Data Hiding in Video,” *Proceedings of IEEE International Conference of Image Processing*, Kobe, Japan, vol. 1, pp. 311-315, Dec 1999.

[2] F. Hartung and B. Girod, “Watermarking of Uncompressed and Compressed Video,” *Signal Processing*, vol. 66, pp. 283-301, 1998.

[3] A. Giannoula and D. Hatzinakos, “Compressive Data Hiding For Video Signals,” *Proceedings of IEEE International Conference of Image Processing*, Barcelona, Spain, vol. 1, pp. 529-532, Sept. 14-17, 2003.

[4] K. F. Chien and W. H. Tsai “Authentication of surveillance video sequences and contents by hiding motion vector information,” *Proceedings of 2006 Conference on Computer Vision, Graphics and Image Processing, Taoyuan, Taiwan, Republic of China*

[5] M. Schneider and S. F. Chang, “A Robust Content Based Digital Signature for Image Authentication,” *Proceedings of IEEE International Conference on Image Processing*, Lausanne, Switzerland, vol. 3, pp. 227-230, Sept. 1996.

[6] D. He et al., “An Object Based Watermarking Solution for MPEG4 Video Authentication,” Proceedings of IEEE International Symposium on Acoustics, Speech, and Signal Processing, Hong Kong, vol. 3, pp. 537-540, Apr. 6-10, 2003.