

Chapter 5

Searches of Motions in Surveillance Videos by Image Watermarking Technique

5.1 Introduction

In this chapter, we describe the method we propose for searching surveillance videos for motions caused by humans or moving objects. The method is useful because motion of an object in a surveillance video is usually regarded as a suspicious activity. In a traditional surveillance system, a magnetic video tape is used to record surveillance videos. Every time we want to check whether a surveillance video has suspicious activities or not, we must rewind the video tape repeatedly. In order to avoid such tedious manipulations on recorded videos, we propose a method for the modern surveillance system by which we can easily check any suspicious image frame through a friendly user interface. The method is based on the application of image watermarking techniques.

In Section 5.1.1, some related problem definitions are given, and the basic idea of the proposed method is described in Section 5.1.2. In Section 5.2, a detailed moving object detection scheme is presented. In Section 5.3, the proposed process for embedding moving object information is stated. In Section 5.4, the proposed process of extracting moving object information is stated. In Section 5.5, experimental results are shown to prove the feasibility of the proposed method. Finally, some discussions and a summary will be made in the last section of this chapter.

5.1.1 Problem Definition

The first issue involved in the proposed method is how to detect moving objects in a real-time surveillance system with a stationary camera. Moving object detection has been a popular topic in computer vision related research for many years. It is often used in surveillance systems for the intrusion detection and can be implemented by various algorithms, such as frame differencing, background modeling, motion vector analysis, etc.

The second issue is how to classify detected moving objects, and a third issue is how to embed the classified results as invisible watermarks into MPEG-4 compressed bitstreams during the encoding process.

5.1.2 Proposed Idea

In the proposed system, each frame captured from a stationary camera is encoded into a compressed bitstream in real time by an MPEG-4 encoder. Before the process of encoding is conducted, a certain moving object detection algorithm is utilized to detect whether there are motions in the frame or not. While motions are detected, we can recognize the types of the detected moving objects by their specific features like human skin color. Finally, the type information is embedded into the quantized frequency domain of an I frame near the frame with motions detected. Therefore, we can easily check whether there are suspicious activities or not in the surveillance video by extracting the embedded information during the decoding process. Moreover, we can even find out what type of moving object is detected in that frame.

5.2 Detection and Classification of Moving Objects

In the proposed system, we utilize an absolute frame differencing algorithm, which compares an input image frame with a fixed background image, to detect motions in an image frame. It may be the most intuitive and fast algorithm for detecting moving objects, particularly when the video camera is static. Then a human detection algorithm is employed to classify moving objects into two types: human and non-human object. An illustration of the proposed process for detection and classification of moving objects is shown in Figure 5.1. In Section 5.2.1, a review of the employed method of human detection is introduced, and a detailed algorithm of moving object detection is described in Section 5.2.2.

5.2.1 Review of Employed Method of Human Detection

Human detection techniques, which are proposed in many researches recently, can be implemented by several approaches, such as the template matching approach, the neural network approach, the color-based approach, and the motion-based approach. In this study, a color-based approach is adopted, which detects human faces using the $YCbCr$ color space. The approach is widely used in several video compression standards. The primary concept of this approach is that each pixel with the skin color will present similar C_b and C_r values. Therefore, we can get a cluster by gathering the skin color statistics from the $YCbCr$ color space.

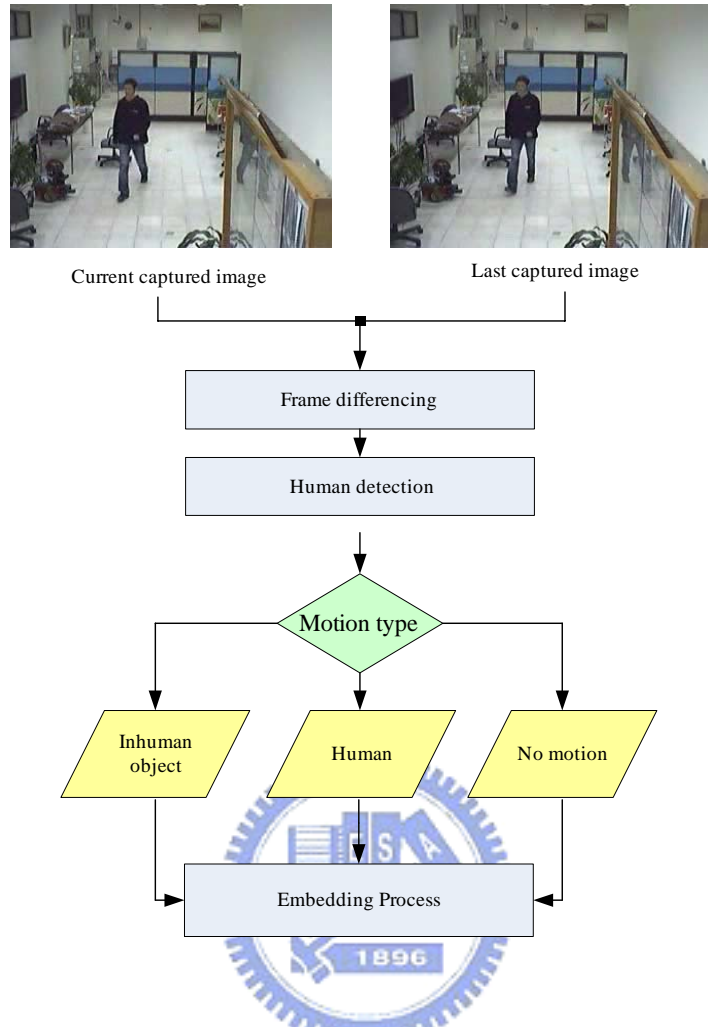


Figure 5.1 Illustration of proposed detection and classification method.

Lee and Yoo [20] proposed an elliptical boundary model for detection of the skin color. In the employed human detection method, we also create a simplified elliptical model presented on a 2D C_b - C_r plane shown in Figure 5.2, where the x -axis represents the C_b value and the y -axis represents the C_r value. For each pixel in the resulting image after performing absolute frame differencing, if its chrominance pair (C_b , C_r) falls within the white elliptical region shown in Figure 5.2, the pixel will be regarded as a skin color pixel. Then we can group such pixels together to form one or more skin color regions.

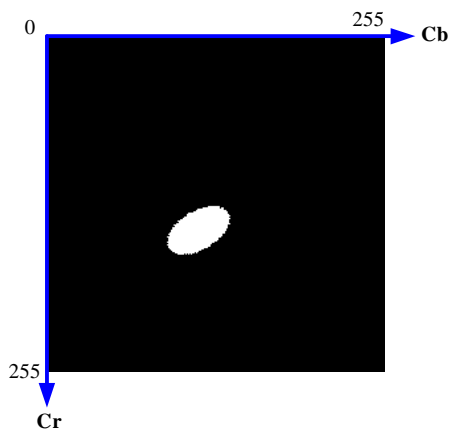


Figure 5.2 Illustration of a binary image representing a simplified elliptical model for skin color detection.

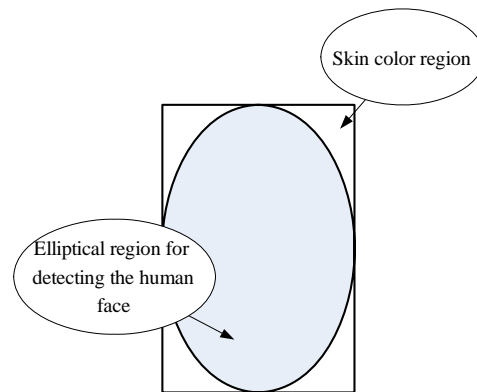


Figure 5.3 Illustration of the elliptical filter for detecting the human face.

Besides, because the shape of the human face is approximately an ellipse, we put an elliptical filter into each detected skin color region and calculate the density of skin-color pixels within the elliptical region shown in Figure 5.3. The region which has the largest density will be considered the right human face region.

5.2.2 Detailed Algorithm

Each captured image is transformed from the RGB into the $YCbCr$ color space in advance. We take the luminance components of each captured image as input to the absolute frame differencing algorithm. Then the current image F is subtracted from the background image F' to get a difference image D . In the following process of human detection, we create a binary image E that represents the simplified skin-color distribution model and use $E(C_b, C_r)$ to denote the binary value of the pixel with the coordinates (C_b, C_r) in E . We obtain the skin color pixels by processing each pixel in D using E and the candidate skin-color regions by applying a region growing algorithm to the identified skin color pixels. The basic concept of the region growing

algorithm in the employed method is to check out eight neighbor pixels N_k around each pixel P_{ij} with the skin color. If the color of each N_k is still similar to the skin color, we check out eight neighbor pixels of each N_k recursively until reaching the boundary of the image frame. A flowchart of the human detection process is shown in Figure 5.4 and a corresponding detailed algorithm is described in the following.

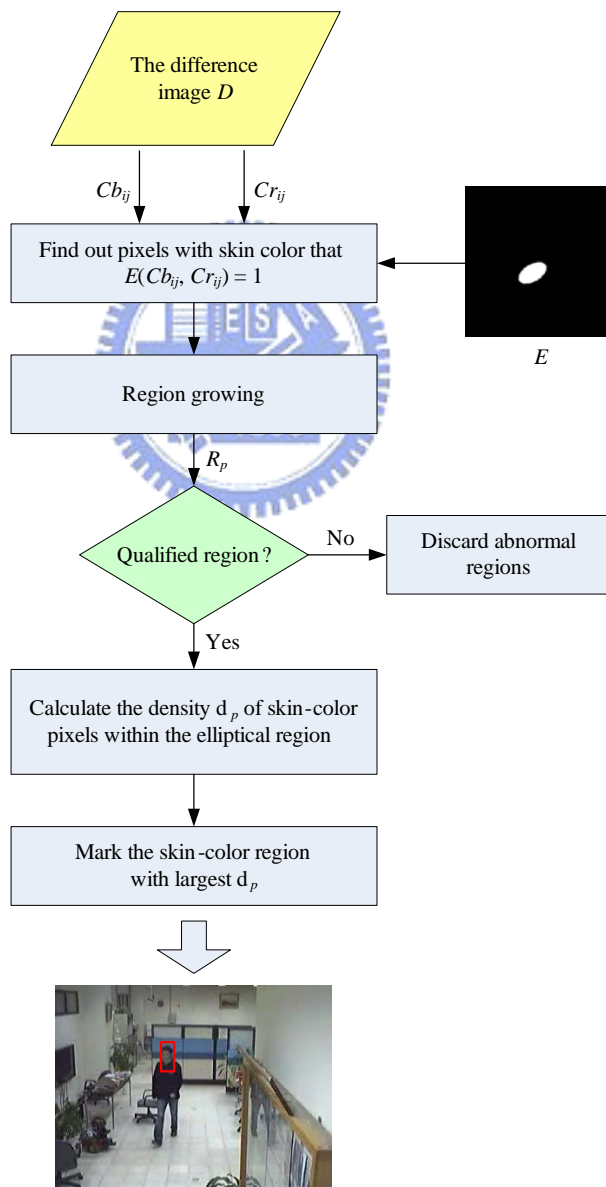


Figure 5.4 Flowchart of the human detection algorithm.

Algorithm 5.1: The process of human detection.


Input: the difference image D in the YC_bC_r color space and a binary image E representing the simplified skin-color distribution model.

Output: the region in D containing a human face.

Steps:

- 1 For each pixel P_{ij} in D , denote its chrominance component as $(C_{b_{ij}}, C_{r_{ij}})$.
- 2 Determine whether the color of P_{ij} is similar to the skin color or not by the following rule:

$$\begin{cases} \text{if } E(C_{b_{ij}}, C_{r_{ij}}) = 1, \text{ then } P_{ij} \text{ is a skin color pixel;} \\ \text{otherwise, } P_{ij} \text{ is not a skin color pixel.} \end{cases} \quad (5.1)$$

- 3 Perform the region growing algorithm to group the skin color pixels in D to form several candidate regions.
- 4 For each candidate skin-color region R , pick out the qualified regions according to the following rules.
 - 4.1 The number of pixels in R must be greater than a pre-defined threshold T_{min} .
 - 4.2 The height H of R must be greater than the width W of R .
 - 4.3 H may not be greater than three times W .
- 5 For each qualified region Q , derive an elliptical region fitting Q and calculate the density d of pixels with the skin-like color within the elliptical region.
- 6 Pick out the region with the largest d as the human face region.

In the above algorithm, the main task of Step 4 is to eliminate regions which are unlikely to be human faces. Such abnormal regions are identified by comparing the axis length ratio of the region, which is defined to be the ratio of the major axis divided by the minor axis, with probable ratios of human faces.

5.3 Real-Time Embedding of Moving Object Information

In this section, we propose a method for embedding moving object information into the I frame during the real-time encoding process. An illustration of the proposed method is shown in Figure 5.5. In Section 5.3.1, the principle of the proposed embedding process is given, and the detailed process of embedding moving object information is described in Section 5.3.2.

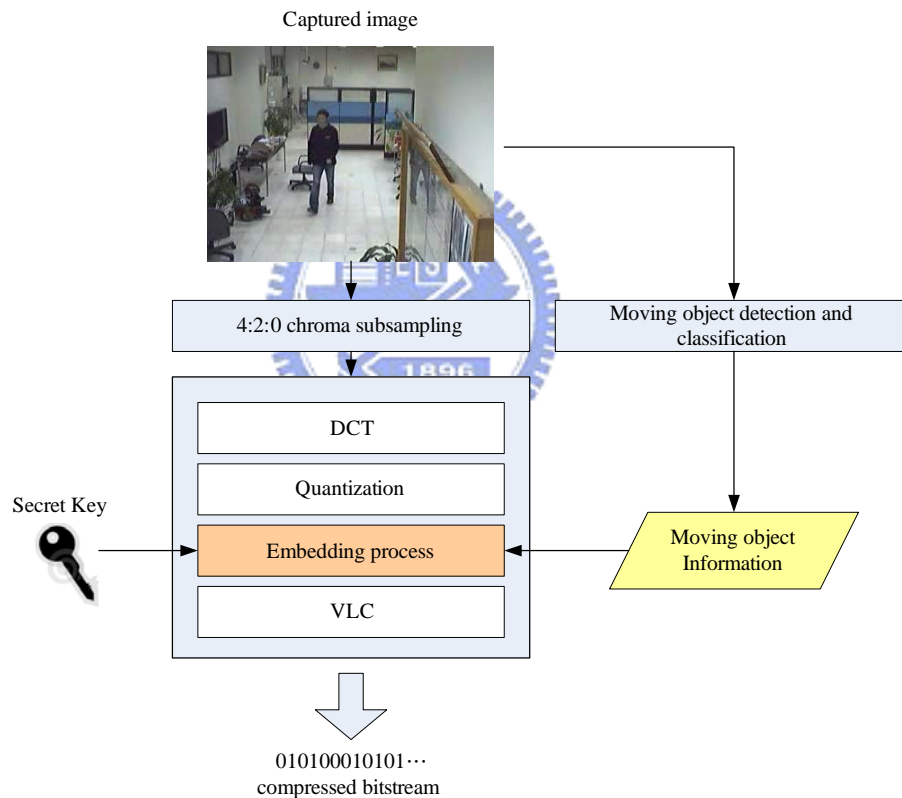


Figure 5.5 Illustration of proposed method for embedding moving object information in real time.

5.3.1 Principle of Proposed Method

In the proposed system, the classified moving object information will be taken as input into the MPEG-4 encoding process while there are motions detected. The input

information is embedded as invisible watermarks to index the surveillance video for the subsequent searching process. If a frame with motions detected is regarded as an I frame by an MPEG-4 encoder, the result of classification will be immediately embedded into the quantized DCT-domain of this I frame; otherwise, the result will be embedded into the next I frame. However, the motion of most moving objects will last for several frames; therefore, the moving object information will be embedded into the I frame only at the beginning of the motion event. That is, motions detected in subsequent frames will be treated as the same kind of moving objects until the end of the motion event.

5.3.2 Proposed Embedding Process

In the proposed embedding process, the 4:2:0 chroma subsampling scheme adopted in the MPEG-4 standard is implemented to subsample each input image before the encoding process. The embedded data can be categorized into three types: moving human, moving object, and no motion. Hence, three different characters are defined as marks for representing respective types of moving objects, and they are combined with the occurring time of motions to form a string I . The string is then transformed into a binary form I_b and each bit of I_b is embedded into an 8×8 luminance block in the quantized DCT-domain.

As mentioned in Chapter 3, we will randomly select one of ten pairs of DCT coefficients defined according to the zig-zag scanning order for embedding one bit of the hidden data by a secret key. A flowchart of the embedding process is illustrated in Figure 5.6 and a detailed algorithm is described in the following.

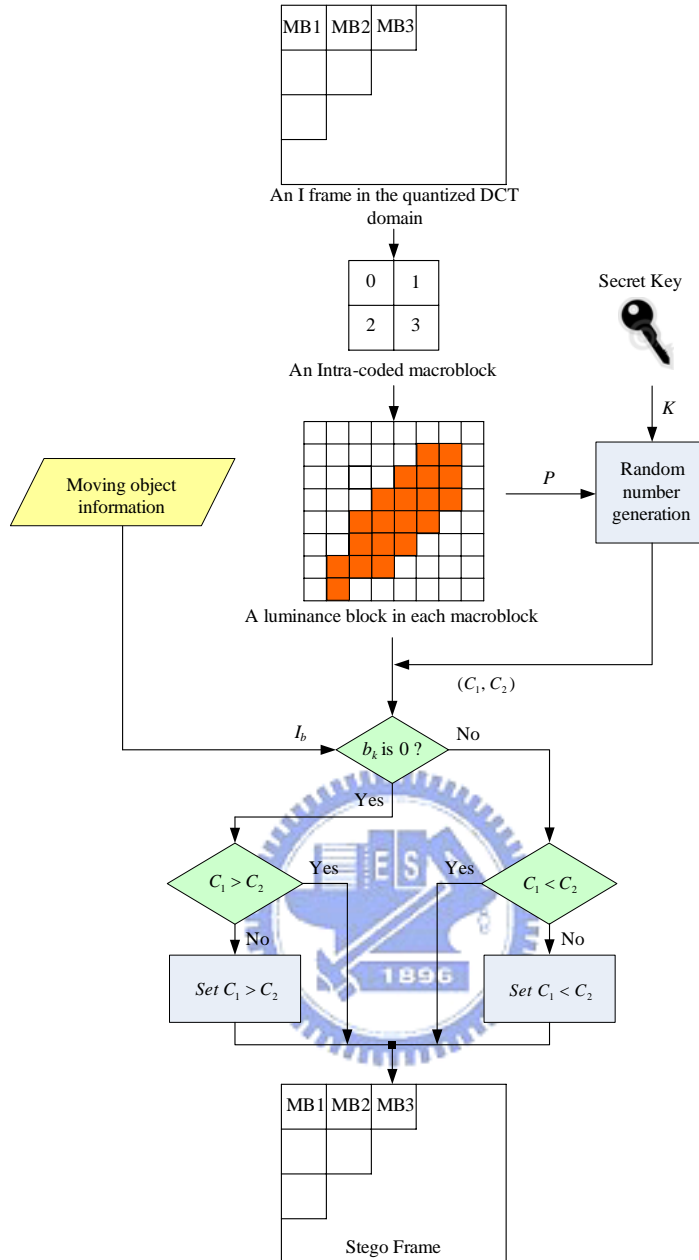


Figure 5.6 Flowchart of process for embedding moving object information.

Algorithm 5.2: The process for embedding moving object information.

Input: an I frame F in the quantized DCT-domain, a secret key K , and moving object information M .

Output: a stego I frame F' .

Steps:

1. Denote the binary form of M as $M_b = b_1b_2b_3\dots b_L$, where L represents the

length of M_b .

2. Combine the input secret key K and the position P of the corresponding 8×8 luminance block B_{ij} to form a seed for a random number generation.
3. Select one pair of DCT coefficients (C_1, C_2) from the ten pre-defined pairs according to the result of random number generation.
4. Embed each bit b_k of M_b by changing the relation between C_1 and C_2 according to the following rule.

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

$$\begin{cases} \text{if } C_1 < C_2, \text{ then swap } C_1 \text{ and } C_2; \\ \text{if } C_1 = C_2, \text{ then set } C_1 = C_2 + T_1, \end{cases} \quad (5.2)$$

where T_1 is a pre-defined threshold.

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

$$\begin{cases} \text{if } C_1 > C_2, \text{ then swap } C_1 \text{ and } C_2; \\ \text{if } C_1 = C_2, \text{ then set } C_2 = C_1 + T_1, \end{cases} \quad (5.3)$$

where T_1 is a pre-defined threshold.

5.4 Extraction of Moving Object Information

In Section 5.4.1, the principle of the proposed extraction process is given and a detailed algorithm of the extraction process is described in Section 5.4.2. An illustration of the proposed searching scheme is shown in Figure 5.7.

5.4.1 Principle of Proposed Method

In the last section, a compressed stego-video is recorded by the proposed system. Before searching motions in the resulting surveillance video, the moving object

information must be extracted from the quantized DCT-domain during the MPEG-4 decoding process. A main advantage of the proposed method is that we can manipulate the encoded information in the compressed domain without incurring the cost of complete decompression. In other words, we can search the classified motions distributed over the surveillance video quickly.

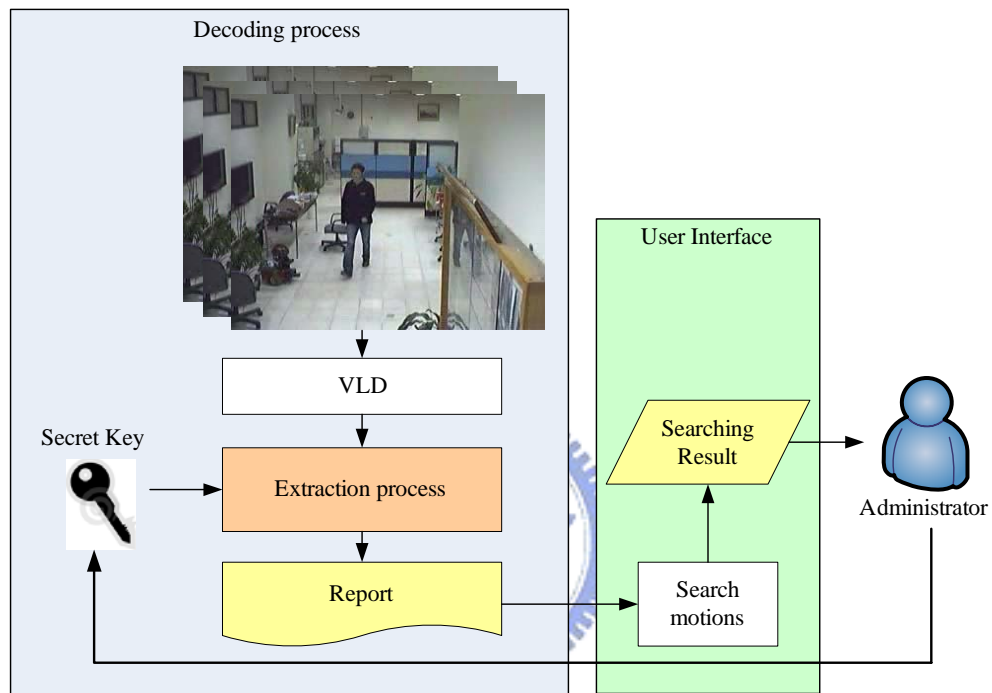


Figure 5.7 Illustration of proposed scheme for searching motions.

5.4.2 Proposed Extraction Process

In the proposed extraction process of moving object information, a variable length decoding process is performed on each I frame to retrieve the quantized DCT coefficients at first. Each bit of hidden data will be extracted from the corresponding 8×8 luminance block. The main task in this process is to check the start mark in the extracted moving object information for recognizing the type of the moving object. A corresponding detailed algorithm is described in the following.

Algorithm 5.3: The process for extracting moving object information.

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

Output: extracted moving object information M' .

Steps:

1. For each 8×8 luminance block B_{ij} with data embedded, combine the input secret key K and the position P of B_{ij} to form a seed for random number generation.
2. Select one pair of DCT coefficients (C_1, C_2) from the ten pre-defined pairs according to the result of random number generation.
3. Extract each bit b_k of M_b' , the binary form of the extracted moving object information, by analyzing the relation between C_1 and C_2 according to the following rule:

When $k \leq L$:

$$\begin{cases} \text{if } C_1 > C_2, \text{ then set } b_k = 0; \\ \text{if } C_1 < C_2, \text{ then set } b_k = 1, \end{cases} \quad (5.4)$$

where L is the length of M_b' .

4. Transform the first eight bits in M_b' into a character C .
5. Check C for classifying the type of the moving object by the following rule:

$$\begin{cases} \text{if } C = "H", \text{ a human in motion is found;} \\ \text{if } C = "O", \text{ an inhuman object in motion is found;} \\ \text{if } C = "N", \text{ no motion is found.} \end{cases} \quad (5.5)$$

6. Transform the rest of bits in M_b' to get the occurring time of the moving object.

5.5 Experimental Results

In our experiments, each image captured by a video camera is encoded in real time by an MPEG-4 encoder to form an MPEG-4 compressed video with frame size 320×240 . We simulate a surveillance system composed of a web camera and a notebook computer. Six frames of the resulting stego-video are shown in Figure. The proposed user interface for searching motions in a stego-video is shown in Figure 5.9 (a). The searching result of a moving person is displayed in Figure 5.9 (b).

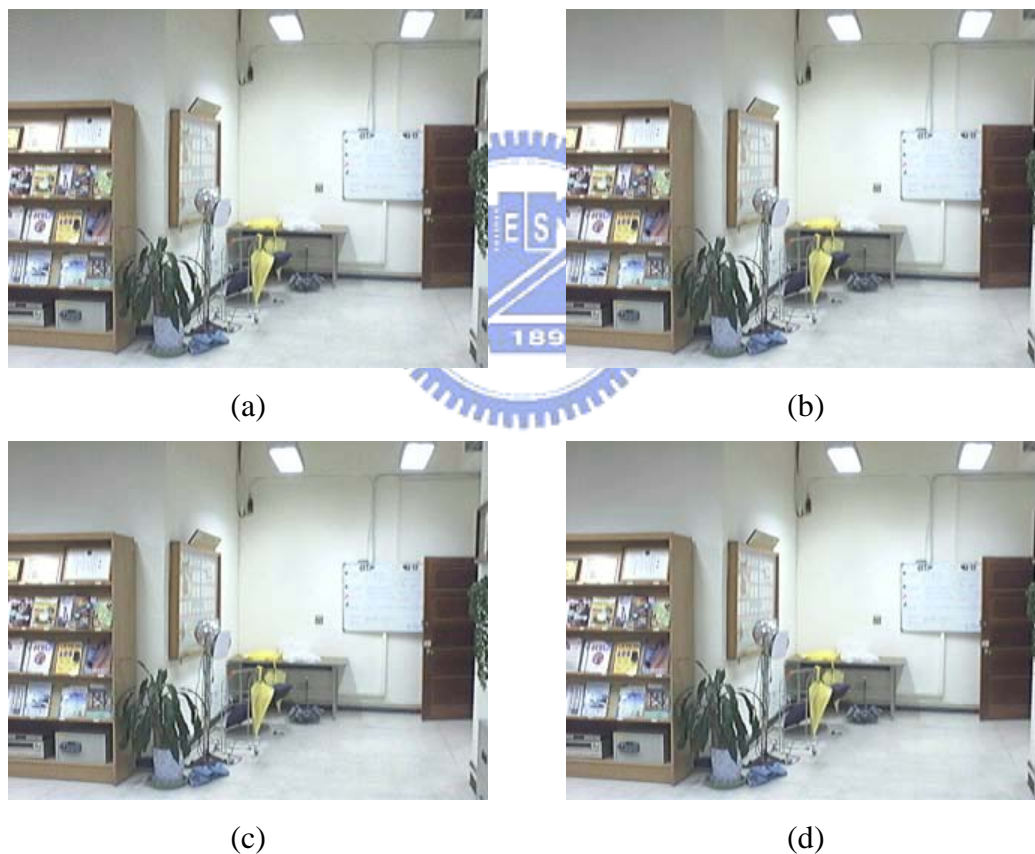


Figure 5.8 Six frames of the resulting stego-video. (a) The first frame (I frame). (b) The second frame (B frame). (c) The third frame (P frame). (d) The 4th frame (B frame). The 5th frame (P frame). The 6th frame (B frame). (continued)

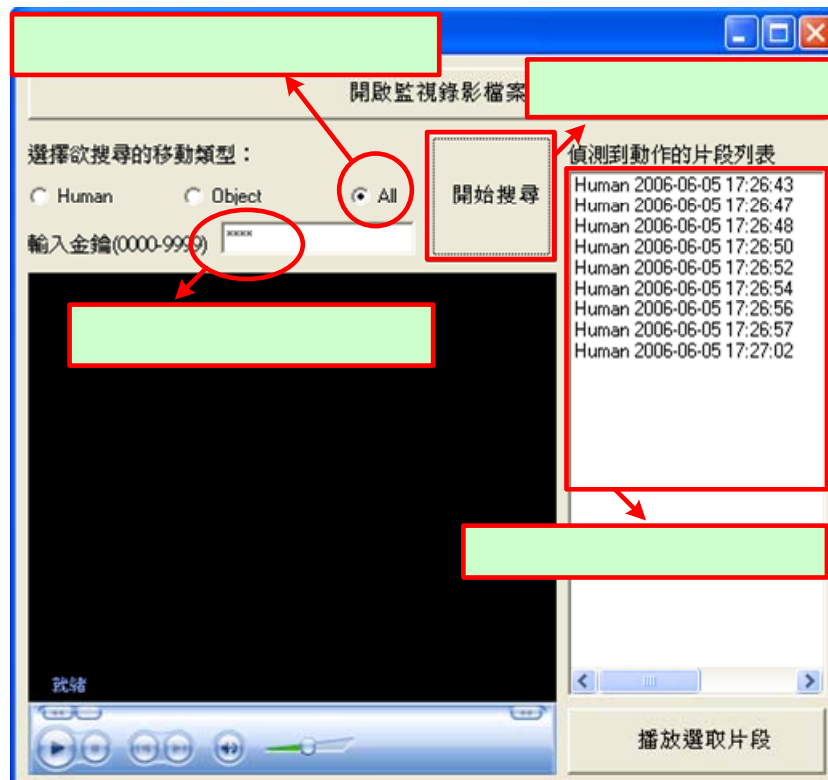


(e)



(f)

Figure 5.8 Six frames of the resulting stego-video. (a) The first frame (I frame). (b) The second frame (B frame). (c) The third frame (P frame). (d) The 4th frame (B frame). The 5th frame (P frame). The 6th frame (B frame). (continued)



(a)

Figure 5.9 (a) The proposed user interface for searching motions. (b) The searching result of a moving person. (continued)

1. Choose the motion type



(b)

Figure 5.9 (a) The proposed user interface for searching motions. (b) The searching result of a moving person. (continued)

5.6 Discussions and Summary

In this chapter, we have proposed a method for a modern surveillance system to search and classify easily and quickly motions occurring in surveillance videos. With the proposed searching method, unlike the traditional one, we can avoid costly overhead caused by completely decompressing videos down to the level of individual frames and by subsequent image processing for detecting moving objects. This advantage comes from the fact that we put the process of detecting moving objects into a real-time MPEG-4 encoding process and embed the detection result into the quantized DCT-domain by the image watermarking technique. However, in order to increase the frame rate of the recorded surveillance video, we need to make some improvements, including increasing the speed of receiving image data from the video

camera and enhancing the computing capability of computers to encode captured images.

