

國立交通大學

資訊科學與工程研究所

碩士論文

利用 WebM 視訊做資訊隱藏及其應用之研究

A Study on Information Hiding Techniques and
Applications via WebM Videos

研究生：曾新翔

指導教授：蔡文祥 教授

中華民國 101 年 6 月

利用 WebM 視訊做資訊隱藏及其應用之研究

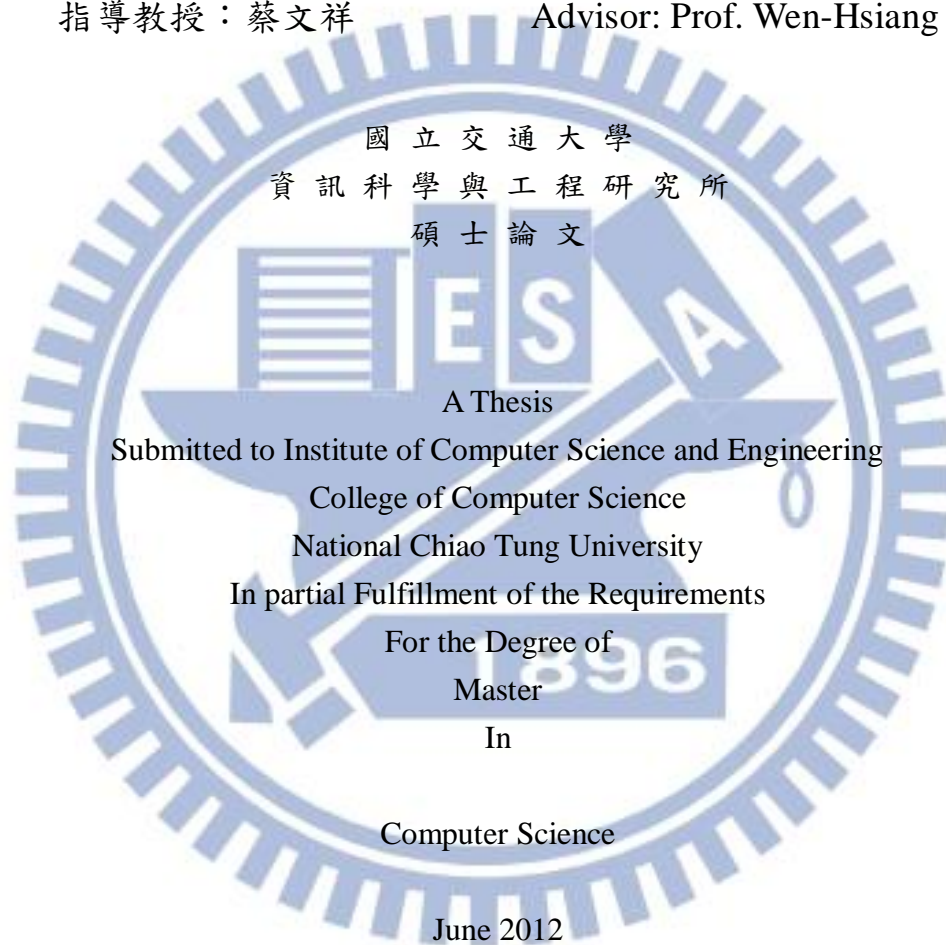
A Study on Information Hiding Techniques and Applications via WebM
Videos

研究生：曾新翔

Student: Hsin-Hsiang Tseng

指導教授：蔡文祥

Advisor: Prof. Wen-Hsiang Tsai



Hsinchu, Taiwan, Republic of China


中華民國 101 年 6 月

利用 WebM 視訊做資訊隱藏及其應用之研究

研究生：曾新翔

指導教授：蔡文祥

國立交通大學資訊科學與工程研究所



摘要

隨著網路以及視訊壓縮技術的進步，數位影片已經成為我們生活中的一部分。WebM 是 Google 公司所開放免版權費用的開源視訊格式，WebM 針對影片在網路上的使用做最佳化。本論文針對 WebM 影片利用資訊隱藏技術做秘密傳輸、視訊驗證及隱私權保護之研究與應用。在秘密傳輸部份，我們提出了一個修改頻率域係數嵌入秘密訊息的方法，不但考慮到秘密訊息的可藏量以及影片嵌入後的品質，也考慮到秘密訊息安全性的問題。在視訊驗證方面，因為視訊監控影片經常成為不法使用者竄改掩蓋犯罪事實的對象，所以我們利用資訊隱藏技術及 WebM 特性，提出一個偵測移動物體作為驗證訊號的方法，對可能遭受竄改的影片做驗證。在隱私權保護部份，因為視訊監控影片經常在個人隱私權方面引起爭議，所以我們提出了一個使用 WebM 特性將具有隱私爭議的影片內容消除及復原的方法。最後我們提出了相關的實驗結果，證明我們所提的方法是可行的。

A Study on Information Hiding Techniques and Applications via WebM Videos

Student: Hsin-Hsiang Tseng

Advisor: Wen-Hsiang Tsai

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

With the advance of the Internet and video compression technologies, uses of digital videos nowadays have become part of the human life. WebM is one of the video coding standards developed in recent years, which has many merits, such as its openness offered by Google, Inc., optimality for uses on the web, etc. In this study, methods for three data hiding applications, namely, covert communication, video authentication, and privacy protection, are proposed using WebM videos as cover media.

For covert communication, a data hiding method via WebM videos by frequency coefficient modifications in the frequency domain is proposed. The method considers not only the data hiding capacity and imperceptibility, but also the security issue.

For video authentication, a method which detects motion objects in a surveillance video to generate authentication signals and embed them in the video to yield a protected version is proposed. The proposed method may be used to verify possible tampering attacks in the protected surveillance video.

For privacy protection in videos, a method for removing privacy-sensitive contents from a video by using WebM features and embedding the removed contents

in the same video imperceptibly is proposed. The hidden privacy-sensitive contents can be extracted later to recover the original privacy-sensitive contents.

Good experimental results show the feasibility of the proposed methods for applications on covert communication, video authentication, and privacy protection in videos.

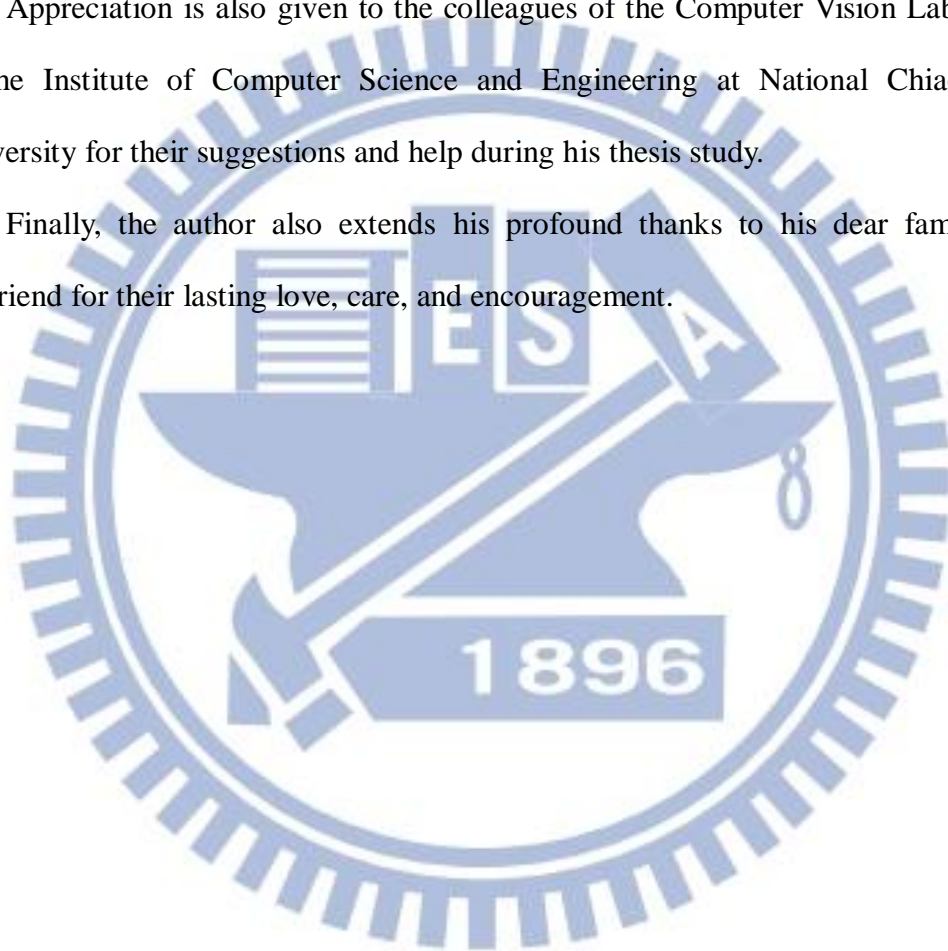


ACKNOWLEDGEMENTS

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

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

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



CONTENTS

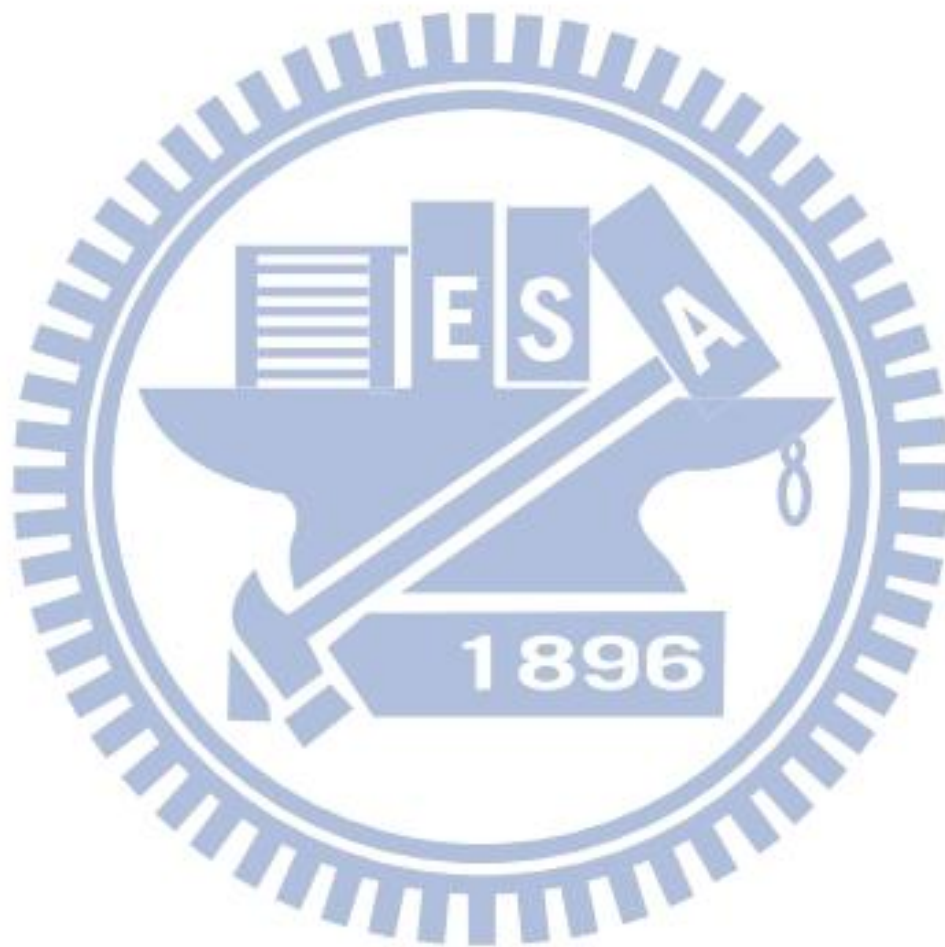
ABSTRACT (in English)	i
ACKNOWLEDGEMENTS	iii
CONTENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	xi

Chapter 1 Introduction	1
1.1 Motivation	1
1.2 General Review of Related Works	3
1.3 Overview of Proposed Methods	3
1.3.1 Terminologies	3
1.3.2 Brief Descriptions of Proposed Methods	4
1.4 Contributions	6
1.5 Thesis Organization	6
Chapter 2 Review of Related Works and WebM Standard	8
2.1 Review of Techniques for Data Hiding via Videos	8
2.2 Review of Techniques for Motion Detection	9
2.3 Review of Techniques for Video Authentication.....	10
2.4 Review of Techniques for Privacy Protection in Videos	11
2.5 Review of WebM Standard	11
2.5.1 Structure of WebM standard.....	12
2.5.2 Process of Encoding.....	13
2.5.3 Process of Decoding	15
2.5.4 Region of Interest maps	17
2.5.5 Reference Frames	18
2.5.6 VP8 Intra Prediction and Inter Prediction	20
Chapter 3 Data Hiding in WebM Videos for Covert Communication by Frequency Coefficient Modifications	24
3.1 Introduction	24
3.1.1 Problem Definition.....	25
3.1.2 Proposed Ideas	25
3.2 Embedding of Secret Data into WebM Videos.....	26
3.2.1 Idea of Proposed Method	26
3.2.2 Process for Embedding Secret Data.....	29

3.3	Extraction of Secret Data from WebM Videos	37
3.4	Experimental Results	38
3.5	Discussions and Summary	44
Chapter 4 Authentication of Surveillance Videos by Motion Object Analysis.....		46
4.1	Introduction	46
4.1.1	Problem Definition.....	47
4.1.2	Proposed Ideas	47
4.2	Generation of Authentication Signals by Motion Contents	48
4.2.1	Principle of Authentication Signal Generation.....	48
4.2.2	Process of Authentication Signal Generation	51
4.3	Embedding and Extracting of Authentication Signals in Surveillance Videos	55
4.3.1	Embedding of Authentication Signals.....	55
4.3.2	Extraction of Authentication Signals	60
4.4	Authentication of Surveillance Videos	65
4.4.1	Detection and Verification of Tampering in Key Frames.....	65
4.4.2	Detection and Verification of Tampering in Prediction Frames	67
4.5	Experimental Results	68
4.6	Discussions and Summary	69
Chapter 5 Protection of Privacy-sensitive Contents in Surveillance Videos Using WebM Video Features		81
5.1	Introduction	81
5.1.1	Problem Definition.....	82
5.1.2	Proposed Ideas	82
5.2	Hiding of Privacy-sensitive Contents	83
5.2.1	Principle of Proposed Method	83
5.2.2	Process for Hiding Privacy-sensitive Contents	85
5.3	Recovery of Privacy-sensitive Contents	91
5.3.1	Proposed Idea	91
5.3.2	Process for Extraction and Recovery Privacy-sensitive Contents.....	91
5.4	Experimental Results	94
5.5	Discussions and Summary	95
Chapter 6 Conslutions and Suggestions for Future Works.....		110

6.1 Conclusions110
6.2 Suggestions for Future Works111

References113



LIST OF FIGURES

Figure 2.1	Flow diagram of WebM encoding process.....	15
Figure 2.2	Top-level hierarchy of WebM video bitstream.....	16
Figure 2.3	Flow diagram of WebM decoding process.....	17
Figure 2.4	an Example of ROI maps of a frame.	17
Figure 2.5	An example of the use of the golden reference frame.	19
Figure 2.6	An example of 4×4 block of pixels.....	21
Figure 2.7	An example of the SPLITMV prediction mode.	23
Figure 2.8	An example of the SPLITMV prediction mode.	23
Figure 3.1	Illustration of the proposed data hiding method.....	26
Figure 3.2	An example of subblocks with yellow coefficients composing a positive-sloped diagonal line.....	28
Figure 3.3	An example of a subblock after performed DCT and quantization with red coefficients composing a positive-sloped diagonal line.....	29
Figure 3.4	The sixteen data patterns for use to embed message data.....	30
Figure 3.5	A flowchart of embedding process.	36
Figure 3.6	A flowchart of proposed message data extraction process.....	39
Figure 3.7	The secret data used in the experiments.....	40
Figure 3.8	The experimental result of computing the PSNR values of chroma frames of tested videos.....	41
Figure 3.9	The 8 th , 9 th , 12 th , and 49 th frames of original video (left) Tempete and stego-video (right).....	42
Figure 3.10	The 24 th and 34 th frames of original video Waterfall (left) and stego-video (right).....	43
Figure 3.11	The 14 th frame of original video Bus (left) and stego-video (right).	43
Figure 3.12	The 27 th frame of original video Container (left) and stego-video (right).	43
Figure 3.13	Extracted secret messages. (a) The correct secret message with a right key. (b) The incorrect secret message with an erroneous key.	44
Figure 4.1	An example of different prediction modes in the content of a video. ...	49
Figure 4.2	An example of motion regions.	50
Figure 4.3	An example of noise macroblocks.....	50
Figure 4.4	The notations of the eight neighboring macroblocks of <i>MB</i>	54
Figure 4.5	A flowchart of the process for embedding authentication signals.....	57
Figure 4.6	A flowchart of the process for authentication signal extraction.	61
Figure 4.7	Six frames of the original video of test1. (a) The 152 th frame (b) The	

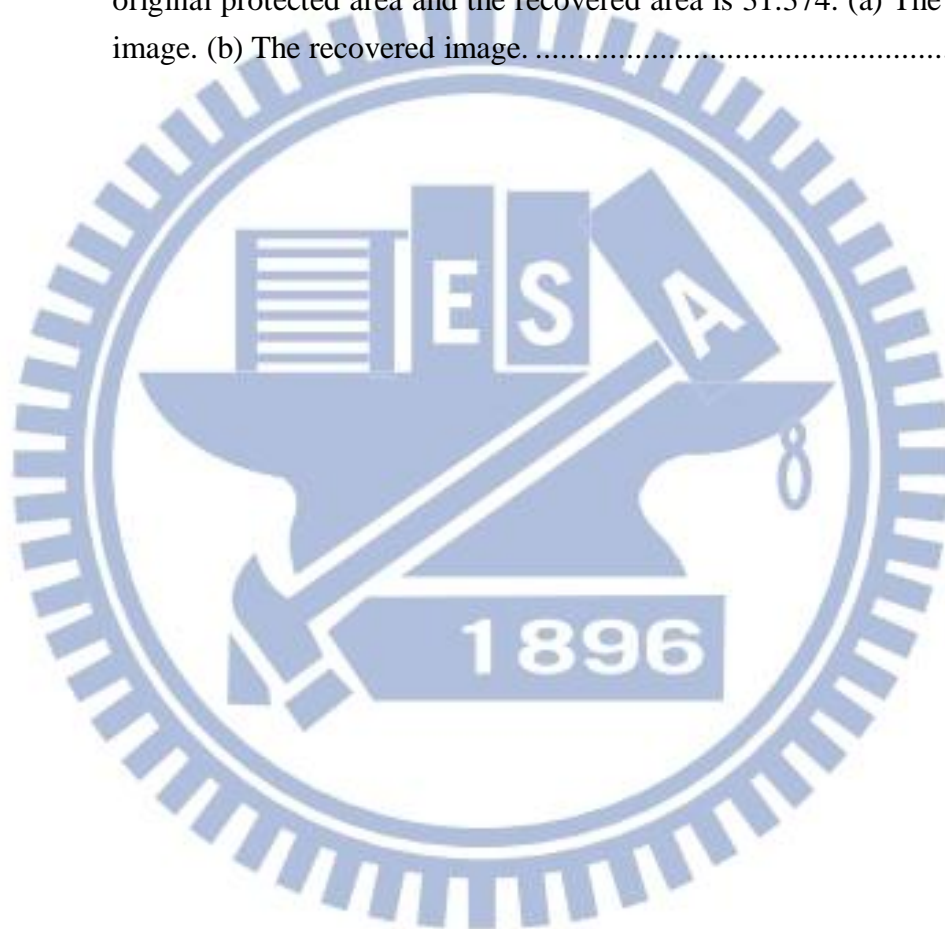
	153 th frame (c) The 154 th frame (d) The 155 th frame (e) The 156 th frame (f) The 157 th frame.....	71
Figure 4.8	Six frames of the protected video of test1. (a) The 152 th frame (b) The 153 th frame (c) The 154 th frame (d) The 155 th frame (e) The 156 th frame (f) The 157 th frame.....	72
Figure 4.9	Six frames of the tampered video of test1. (a) The 152 th frame (b) The 153 th frame (c) The 154 th frame (d) The 155 th frame (e) The 156 th frame (f) The 157 th frame.....	73
Figure 4.10	Six frames of the authenticated video of test1. (a) The 152 th frame (b) The 153 th frame (c) The 154 th frame (d) The 155 th frame (e) The 156 th frame (f) The 157 th frame.....	74
Figure 4.11	Six frames of the protected video of test2. (a) The 48 th frame (b) The 49 th frame (c) The 50 th frame (d) The 51 th frame (e) The 52 th frame (f) The 53 th frame.....	75
Figure 4.12	Six frames of the authenticated video of test2. (a) The 48 th frame (b) The 49 th frame (c) The 50 th frame (d) The 51 th frame (e) The 52 th frame (f) The 53 th frame.....	76
Figure 4.13	Six frames of the protected video of test3. (a) The 68 th frame (b) The 69 th frame (c) The 70 th frame (d) The 71 th frame (e) The 72 th frame (f) The 73 th frame.....	77
Figure 4.14	Six frames of the authenticated video of test3. (a) The 68 th frame (b) The 69 th frame (c) The 70 th frame (d) The 71 th frame (e) The 72 th frame (f) The 73 th frame.....	78
Figure 4.15	Six frames of the protected video of test4. (a) The 141 th frame (b) The 142 th frame (c) The 143 th frame (d) The 144 th frame (e) The 145 th frame (f) The 146 th frame.....	79
Figure 4.16	Six frames of the authenticated video of test4. (a) The 141 th frame (b) The 142 th frame (c) The 143 th frame (d) The 144 th frame (e) The 145 th frame (f) The 146 th frame.....	80
Figure 5.1	An example of reference error.....	84
Figure 5.2	An example of modify the frequency coefficients in an intra-coded macroblock to yield intra-coded grey macroblocks.....	85
Figure 5.3	Comparison between the original image and the image whose Y2 coefficients have been lost. (Left) the original image. (Right) the image whose Y2 coefficients have been lost.	88
Figure 5.4	The zig-zag scan order.	88
Figure 5.5	Six representative frames of a video. (a) The 41 frame. (b) The 53 frame. (c) The 77 frame. (d) The 99 frame. (e) The 124 frame. (f) The 239	

	frame.	96
Figure 5.6	Six representative frames of the protected video of Fig. 5.5. (a) The 41 frame. (b) The 53 frame. (c) The 77 frame. (d) The 99 frame. (e) The 124 frame. (f) The 239 frame.	97
Figure 5.7	Six representative frames of a recovered video. (a) The 41 frame. (b) The 53 frame. (c) The 77 frame. (d) The 99 frame. (e) The 124 frame. (f) The 239 frame.....	98
Figure 5.8	Six frames of a second video. (a) The 40 th frame (b) The 41 th frame (c) The 42 th frame (d) The 43 th frame (e) The 44 th frame (f) The 45 th frame.	99
Figure 5.9	Six frames of the protected video of Fig. 5.8. (a) The 40 th frame (b) The 41 th frame (c) The 42 th frame (d) The 43 th frame (e) The 44 th frame (f) The 45 th frame.	100
Figure 5.10	Six frames of the recovered video. (a) The 40 th frame (b) The 41 th frame (c) The 42 th frame (d) The 43 th frame (e) The 44 th frame (f) The 45 th frame.	101
Figure 5.11	Six frames of a third video. (a) The 131 th frame (b) The 137 th frame (c) The 147 th frame (d) The 160 th frame (e) The 174 th frame (f) The 181 th frame.	102
Figure 5.12	Six frames of the protected video of Fig. 5.11. (a) The 131 th frame (b) The 137 th frame (c) The 147 th frame (d) The 160 th frame (e) The 174 th frame (f) The 181 th frame.	103
Figure 5.13	Six frames of the recovered video. (a) The 131 th frame (b) The 137 th frame (c) The 147 th frame (d) The 160 th frame (e) The 174 th frame (f) The 181 th frame.	104
Figure 5.14	Six frames of a fourth video. (a) The 195 th frame (b) The 209 th frame (c) The 223 th frame (d) The 236 th frame (e) The 246 th frame (f) The 255 th frame.	105
Figure 5.15	Six frames of the protected video of Fig. 5.14. (a) The 195 th frame (b) The 209 th frame (c) The 223 th frame (d) The 236 th frame (e) The 246 th frame (f) The 255 th frame.	106
Figure 5.16	Six frames of the recovered video. (a) The 195 th frame (b) The 209 th frame (c) The 223 th frame (d) The 236 th frame (e) The 246 th frame (f) The 255 th frame.	107
Figure 5.17	Comparison between an original image and the corresponding recovered image. (The 77 th frame) The average value of PSNR of the recovered area with respect to between the original protected area is 35.73. (a) The original image. (b) The recovered image.	108
Figure 5.18	Comparison between an original image and the corresponding recovered	

image.(The 43th frame) The average value of PSNR of the recovered area with respect to between the original protected area is 30.372. (a) The original image. (b) The recovered image.108

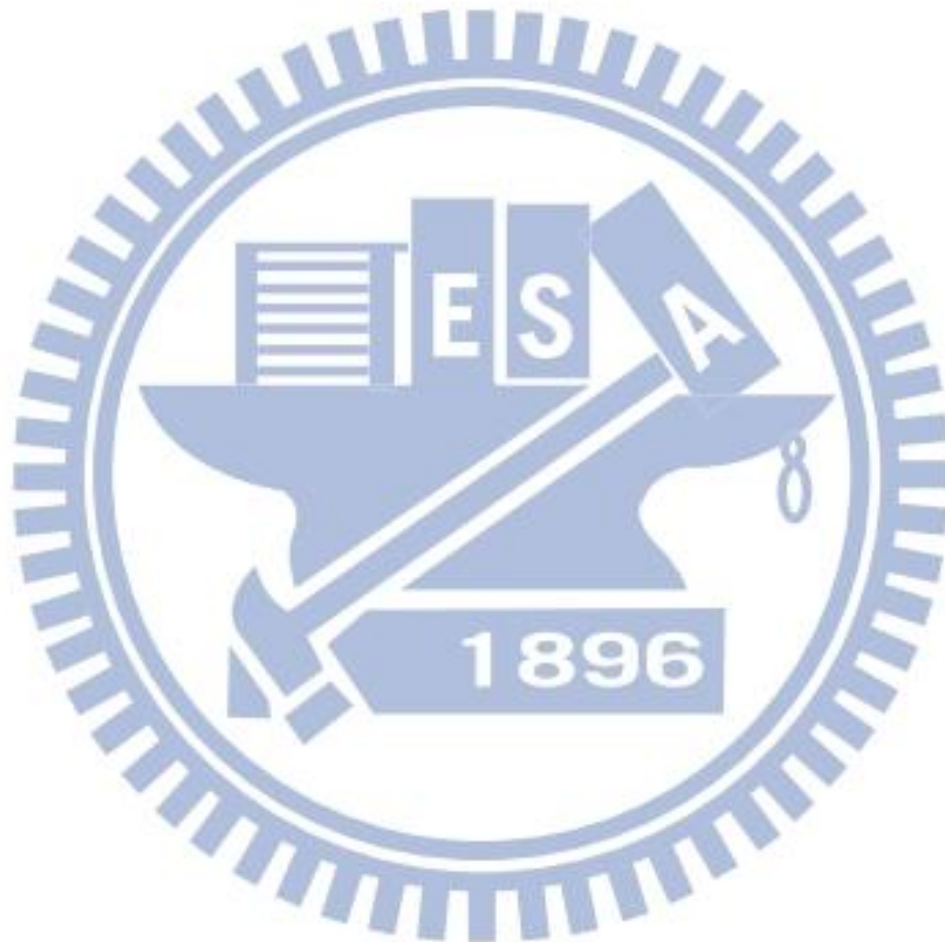
Figure 5.19 Comparison between an original image and the corresponding recovered image.(The 147th frame) The average value of PSNR of the recovered area with respect to between the original protected area is 33.361. (a) The original image. (b) The recovered image.108

Figure 5.20 Comparison between an original image and the corresponding recovered image.(The 246th frame) The average value of PSNR between the original protected area and the recovered area is 31.374. (a) The original image. (b) The recovered image.109



LIST OF TABLES

- Table 3.1 Configuration parameters used in this study. 錯誤! 尚未定義書籤。
- Table 3.2 Values of NHB, PSNRI, and VSI of several video clips.錯誤! 尚未定義書籤。
- Table 4.1 Experimental results of loss rate of authentication signals of test1 video (total number of frames in test1 video is 270, and total number of authentication signals in test1 video is 87)..... 錯誤! 尚未定義書籤。



Chapter 1

Introduction

1.1 Motivation

With the advance of the Internet and video compression technologies, uses of digital videos nowadays have become part of the human life. WebM is one of the video coding standards developed in recent years, which contains many merits, such as its openness offered by Google Inc., optimality for uses on the web, etc. Specifically, it is well known that the success of the web technology comes mainly from some core techniques such as HTML, HTTP, and TCP/IP whose formats were designed to be open for people to implement and improve. Similarly, as the video is a core to the web experience, WebM was also designed to be 100% free and open-sourced. Also, the WebM format is defined to optimal for the web, in the sense of enabling convenient playbacks on any device, including low-power netbooks, handhelds, tablets, etc. Because of the efficiency and good quality of the WebM video, some popular video sharing web sites, like YouTube, have already used WebM videos widely for user communications. So, WebM is considered very suitable for use as a kind of carrier for information hiding and is investigated in depth in this study.

From another point of view, data hiding techniques can be used to hide *secret messages* into given *cover videos*, resulting in so-called *stego-videos*. In this way, stego-videos instead of secret messages themselves may be transmitted through networks. Except the sender of the stego-image and authorized receivers, other users do not know the existence of the hidden information and so will not try to “dig” the

information inside because the secret data hidden in a stego-video are usually invisible. It is desired in this study to develop data hiding methods via WebM videos for covert communication.

Another issue is that nowadays video surveillance via uses of video cameras is *everywhere* around our living spaces. In some cities like London, thousands or even more cameras have been deployed around every corner of the cities. Video fidelity and privacy so arouse serious concerns — because surveillance videos may contain suspicious or unlawful actions, malicious users might try to acquire videos in illegal ways and tamper with them for misrepresentation. Consequently, we have to consider the fidelity issue of surveillance videos. In addition, a surveillance camera may monitor both public and private spaces in the meantime, violating possibly personal privacy by acquiring images of private individuals' activities. Therefore, it is necessary to develop effective methods for authenticating and protecting surveillance videos of the WebM type.

As mentioned above, if surveillance videos contain suspicious or unlawful actions, malicious users might try to acquire videos in illegal ways and tamper with them for misrepresentation. So, *video authentication* is essential in video surveillance applications and has become a main topic for researches. Embedding invisible authentication signals in a video, which results in a protected video, is a good approach to video authentication. If a malicious user tampers with a protected video, the authentication signal hidden in the video can used to detect and display the tampering. How to generate authentication signals and verify protected surveillance videos are also goals of this study.

Besides, privacy protection is a very important issue in video surveillance. Since video surveillance systems exist everywhere in our environment and usually conduct space monitoring for long time periods, it may record information of individuals and

so violate protection of personal privacy. Hence, it is necessary in some cases to hide the privacy-violating parts of the surveillance video content to avoid legal disputes and to protect personal privacy from being misused by suspicious people. This is the final goal of this study.

1.2 General Review of Related Works

Although all the above-mentioned goals of this study are related to the technique of information hiding via videos, different methods should be adopted for different applications. For motion detection, many motion detection algorithms such as temporal differencing, background subtraction, etc. used to detect moving objects have been proposed. For video data hiding, techniques like hiding in motion vectors of the video data, modulating the partition size which is a feature of the video format, etc., have been proposed. For video authentication, techniques like watermarking, digital signatures, etc. are widely used. For privacy protection, lots of techniques have been proposed, such as scrambling of image areas which contain sensitive information, using a set of visual abstraction operators such as silhouette and transparency which controlling the disclosure of individuals' privacy visual information, etc. In addition, because the proposed data hiding, authentication, and privacy protection techniques are applied to WebM videos, we will also make a review of the WebM standard in Chapter 2.

1.3 Overview of Proposed Methods

1.3.1 Terminologies

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

follows.

1. *Secret*: a secret is a piece of information that is important and should be preserved properly and not revealed to unauthorized people.
2. *Stego-video*: a stego-video is one in which some digital message data are embedded.
3. *Motion region*: a motion region is an area containing motion objects in an input video after a motion detection process.
4. *Protected video*: a protected video is one into which authentication signals have been embedded.
5. *Video authentication*: video authentication is a process for verifying the integrity and fidelity of a protected surveillance video by checking the authentication signals embedded in it.
6. *Privacy-protected area*: a privacy-protected area is part of the content of a surveillance video, in which privacy-violating information has been removed to avoid legal disputes and to protect personal privacy from being misused by suspicious people.
7. *Recovered privacy-protected area*: a recovered privacy-protected area is part of the content of a video which, being removed before due to privacy-violation, is recovered from a privacy-protected area defined above.

1.3.2 Brief Descriptions of Proposed Methods

In this study, we have developed several methods for data hiding and its applications via WebM videos. They are briefly described in the following.

(A) *A data hiding method for covert communication by modifying frequency coefficients in WebM video data —*

A data hiding method which modifies the frequency coefficients of WebM video data for covert communication is proposed in this study. Because the VP8 video codec always conducts compression transformations at the 4×4 resolution, the proposed method modifies the WebM's frequency coefficients of the chroma color channels for data hiding. In addition, the PSNR values are computed and compared with a threshold to optimize these changes for maintaining the video quality. For the secret security issue, we select randomly data hiding positions in images in order to reduce the possibility for a malicious user to figure out the locations where the secret data are embedded.

(B) A method for authentication of surveillance videos by analyzing motion objects —

A method using the prediction-mode information and motion vectors in the VP8 video codec to detect motion objects and group them into motion regions to generate authentication signals is proposed for video authentication in this study. If a protected video has been tampered with, according to the authentication signals embedded in it, the proposed authentication system can detect and verify this video to indicate how and where the protected video is tampered with.

(C) A method for protection of personal privacy in surveillance videos using WebM video features —

A method using an information hiding technique is proposed in this study for removing, hiding, and recovering video contents containing sensitive personal information. A video can be decoded correctly based on some decoding information including motion vectors and frequency coefficients. Therefore, the original decoding information may be removed from the original video stream and set to some predefined values in order to cover video contents with sensitive

privacy information and replace them with background image parts. The removed decoding information is not eliminated but embedded into the video and can be extracted later from the privacy protected video in case there is a need of retrieving the privacy-sensitive contents.

1.4 Contributions

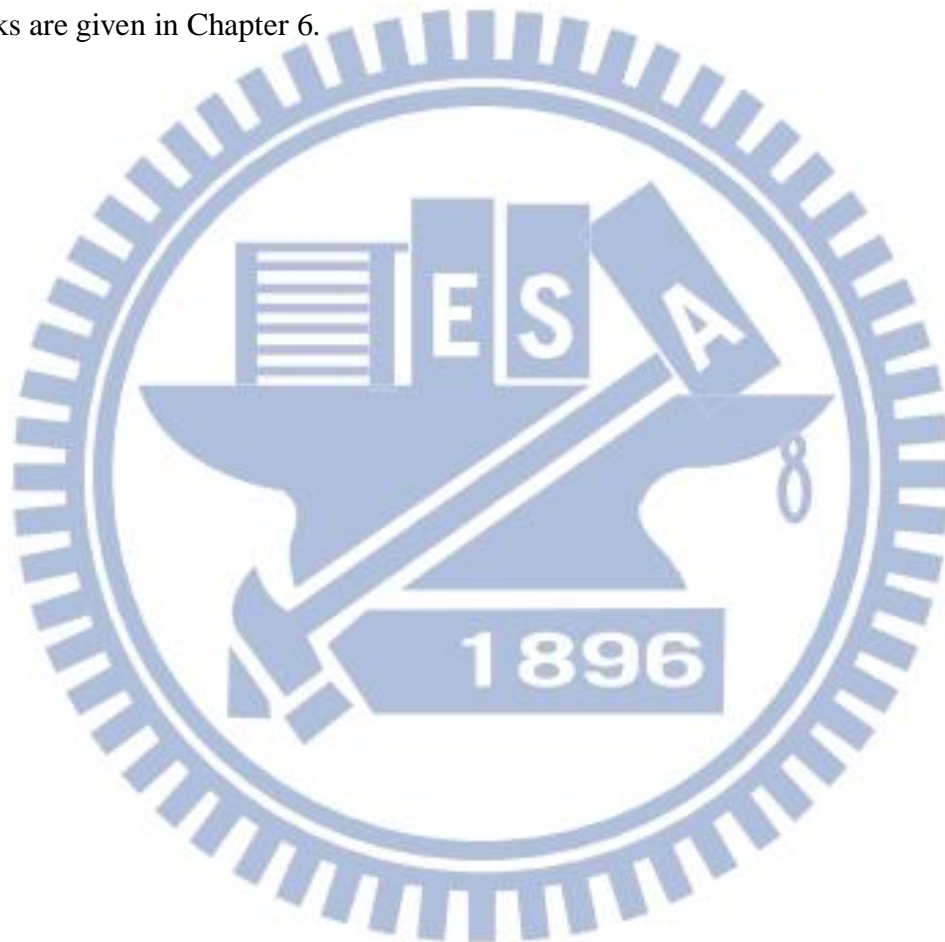
New methods for data hiding and applications via WebM videos are proposed in this study. The contributions made in this study are summarized in the following.

1. For the first time, WebM videos are used as carriers for information hiding.
2. New applications of using the region-of-interest map, which is a feature of the VP8 video codec in the WebM video, are proposed.
3. A data hiding method based on some properties of the VP8 video codec is proposed.
4. A method of motion detection in videos based on the prediction-mode information in the VP8 video codec is proposed.
5. A video authentication system using some features of the VP8 video codec as authentication signals is proposed.
6. A new application of using the golden reference frame in the VP8 video codec to solve the privacy protection problem is proposed.
7. A method for hiding the privacy-sensitive content in a given WebM video and restoring it later is proposed.

1.5 Thesis Organization

In the remainder of this thesis, a detailed review of related works about motion

detection, video data hiding, video authentication, and privacy protection in surveillance videos, as well as the WebM standard is given in Chapter 2. In Chapter 3, the proposed method for data hiding via WebM videos for covert communication is described. In Chapter 4, the proposed video authentication system for surveillance videos is described. In Chapter 5, the proposed method of privacy protection of surveillance videos is presented. Finally, conclusions and some suggestions for future works are given in Chapter 6.



Chapter 2

Review of Related Works and WebM Standard

In this chapter, we give a survey of related works about data hiding, motion detection, video authentication, and privacy protection in videos in Sections 2.1 through 2.4, respectively. Then, we give a review of the standard of the WebM video in Section 2.5.

2.1 Review of Techniques for Data Hiding via Videos

Lots of data hiding Techniques have been developed for hiding secret data into various media and documents in the past decade. By this way, secret data can be transmitted covertly or kept securely for various applications. Because the capacities of hiding data in videos are usually larger than hiding data in images or documents, many data hiding techniques via videos have been proposed [1-4]. Hu et al. [1] proposed a method for hiding data in H.264/AVC videos based on the use of intra-prediction modes. The basic idea is to modify 4×4 intra-prediction modes based on a mapping between 4×4 intra-modes and hidden bits. Their method uses only the intra-coded macroblock to hide data. Hussein [2] proposed a method for embedding data in motion vectors based on their associated prediction error. Yang and Bourbakis [3] proposed a method for embedding data in the DCT coefficients by means of vector

quantization. Kapotas et al. [4] proposed a method for embedding data into encoded video sequences, in which the hiding technique is used to modulate the partition size to hide the secret data. This method can only be used for embedding information in inter-coded macroblocks.

2.2 Review of Techniques for Motion Detection

A lot of motion detection techniques have been proposed to detect moving objects in videos [5-9]. The techniques can be classified into two categories. One is for use in the pixel domain [5-6] and the other in the compressed domain [7-9]. 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, such as H.264/AVC or WebM.

Specifically, Haritaoglu et al. [5] proposed a motion detection method based on background subtraction in the pixel domain. They built a statistical model for a background scene that allows them to detect moving objects even when the background scene is not completely stationary. Lipton et al. [6] proposed another approach based on temporal differencing in the pixel domain, which computes pixel-wise differences between consecutive video frames separated by a constant time to find moving objects. Zeng et al. [7] proposed another approach in the compressed domain by employing a block-based Markov random field (MRF) model in a field formed with motion vectors to segment moving objects during a decoding process.

Babu et al. [8] proposed an automatic video object segmentation algorithm for the MPEG video. They estimated first the number of independently moving objects in the scene using a block-based affine clustering method. Object segmentation is then accomplished by an expectation maximization (EM) clustering algorithm. Spyridon et al. [9] proposed a method for automatic direct detection of moving objects in the H.264 compressed domain. Different blocks/sub-blocks are combined with their associated motion vectors in order to denote a moving object. Their method works in the compressed domain as the block-sizes and the motion vectors can be found by partially decoding the H.264 bitstream.

2.3 Review of Techniques for Video Authentication

Video authentication plays an important role in a digital-rights-management system, so many different methods have been proposed to solve the problem [10-12]. Zhang and Ho [10] introduced a video authentication method which makes an accurate usage of tree-structured motion compensation, motion estimation, and Lagrange optimization of the H.264 standard. As mentioned in the paper, authentication information is embedded according to a best-mode decision strategy in the sense that if a video undergoes any spatial and temporal attacks, the scheme can detect the tampering by the sensitive mode change. Pröfrock et al. [11] proposed a method using skipped macroblocks of an H.264 video to embed authentication data. The data are embedded as a fragile, blind, and erasable watermark with low video quality degradations. In contrast with other authentication methods, the embedding process is done after an H.264 compression process, while others are done during the process. The methods mentioned above usually use additional authentication

information to authenticate videos. Ait Saadi et al. [12] proposed a method using content based digital signatures from the transform domain as fragile watermarks and then embeds them in motion vectors with the best partition mode in tree-structured motion compensation.

2.4 Review of Techniques for Privacy Protection in Videos

Privacy protection has become an important issue along with video surveillance systems. Many different approaches have been introduced in recent years [13-16]. Dufaux et al. [13] introduced a method to protect personal privacy by scrambling regions containing personal information. As a consequence, the scene remains visible, but the privacy-sensitive information is not identifiable. Meuel et al. [14] introduced a method to protect faces in surveillance videos. Any visible information of faces in a video is deleted and embedded in the video that allows further reconstruction of the faces if needed. Zhang et al. [15] proposed a method to protect authorized persons, which are not only removed from a surveillance video, but also embedded into the video. Yu et al. [16] proposed another method protecting individuals' privacy by controlling the disclosure of individuals' private visual information. A set of visual abstraction operators such as silhouette and transparency is applied, which gradually control individuals' private visual information.

2.5 Review of WebM Standard

In this study, all the proposed information hiding, video authentication, and privacy protection techniques employ WebM videos as carriers for hiding information.

The WebM project, which is a project founded by Google Inc., is aimed to describe the detail for the WebM standard, which can be found at the WebM project website [17]. We give a brief review of the WebM standard in this section. In Section 2.5.1, the structure of the WebM standard will be described. In Sections 2.5.2 and 2.5.3, the encoding and decoding processes in the WebM standard are described, respectively. In Sections 2.5.4, 2.5.5, and 2.5.6, related WebM features are described.

2.5.1 Structure of WebM standard

WebM is an open media file format designed for the web whose openness was offered by Google Inc. in May 2010. Each WebM file consists of video streams compressed with the VP8 video codec and audio streams compressed with the Vorbis audio codec. The WebM file structure is based on the Matroska media container. All of them are royalty-free patent license products, so developers could develop or do researches on them without considering any patent suit issue.

The VP8 video codec works exclusively with an 8-bit YUV 4:2:0 image format, each 8-bit chroma pixel in the two chroma color space (U and V) corresponds to a 2x2 block of 8-bit luma pixels in the luma color space (Y), and the coordinates of the upper left corner of the Y block are exactly twice the coordinates of the corresponding chroma pixels. The pixels are simply a large array of bytes stored in rows from top to bottom, each row being stored from left to right. This “left to right” then “top to bottom” raster-scan order is reflected in the layout of the compressed data.

Also, each frame is decomposed into an array of macroblocks. A macroblock is a square array of pixels whose Y dimensions are 16x16 and whose U and V dimensions are 8x8. The macroblock-level data in a compressed frame are also processed in a raster-scan order. The macroblocks are further decomposed into 4x4 subblocks. So

every macroblock has sixteen Y subblocks, four U subblocks, and four V subblocks.

Like other video codecs, the VP8 video codec also has a transform process which converts pixels in the spatial domain into coefficients in the frequency domain. In the VP8 video codec, the discrete cosine transform (DCT) and the Walsh-Hadamard transform (WHT) always conduct compression at the 4×4 resolution. The DCT is used for the sixteen Y, four U, and four V subblocks. The WHT is used to encode a 4×4 array comprising the average intensities of the sixteen Y subblocks of a macroblock. These average intensities are, up to a constant normalization factor, nothing more than the zeroth DCT coefficients of the Y subblocks. The VP8 video codec considers this 4×4 array as a second-order subblock called Y2.

There are two frame types in the VP8 video codec which are *intra-frame* and *inter-frame*. Intra-frames (also called key frames or I-frames) are decoded without reference to any other frame in a sequence. Key frames provide random access points in a video stream. Inter-frames (also called *prediction frames* or *P-frames*) are encoded with reference to prior frames, specifically all prior frames up to and including the most recent key frame. The VP8 video codec uses three types of reference frames for prediction frames: *prior frame*, *golden reference frame*, and *alternate reference frame*. We will have more illustrations about the golden reference frame and the alternate reference frame which are features of the VP8 video codec in Section 2.5.5.

2.5.2 Process of Encoding

The process of encoding of WebM videos is illustrated in Figure 2.1. There are two data flow paths, forward and reconstruction. In the forward path, a macroblock is

encoded in the intra-mode or inter-mode. In the intra-mode, the encoder calculates the best intra-prediction mode which uses the current encoded blocks as references. In the inter-mode, the encoder calculates the best inter-prediction mode from the last frame or the golden reference frame. After deciding the prediction mode, the encoder generates prediction blocks/buffers. In the intra-mode, the encoder subtracts 128 from each pixel which needs to be encoded. In the inter-mode, the encoder subtracts values of pixels of the current block from those of corresponding pixels of a block which is selected by the motion vector. Both the intra-mode and the inter-mode will produce a residual block.

Also, each 16×16 macroblock is divided into sixteen 4×4 DCT blocks, each of which is transformed by a bit-exact DCT approximation. After the DC coefficients of these bit-exact DCT blocks are collected into another group, all DC coefficients set as zero. Furthermore, this group performs the Walsh-Hadamard transform in order to increase the compression rate. After that, transformed coefficients of these blocks are quantized. Then, each resulting block is scanned in a zig-zag order and entropy encoded. Here, entropy coding is the process of taking all information from all the other processes: DCT coefficients, prediction mode, motion vectors, and so forth — and compressing them losslessly into the final output file.

In the reconstruction path, the encoder decodes (reconstructs) each block in a macroblock which is regarded as a reference for further prediction. The quantized coefficients are scaled and inverse-transformed to product a difference block, and then the prediction is added to the difference block to product a reconstructed block. Finally, a loop filter is used to reduce the effects of blocking distortion and the reconstructed reference picture is created from a series of blocks.

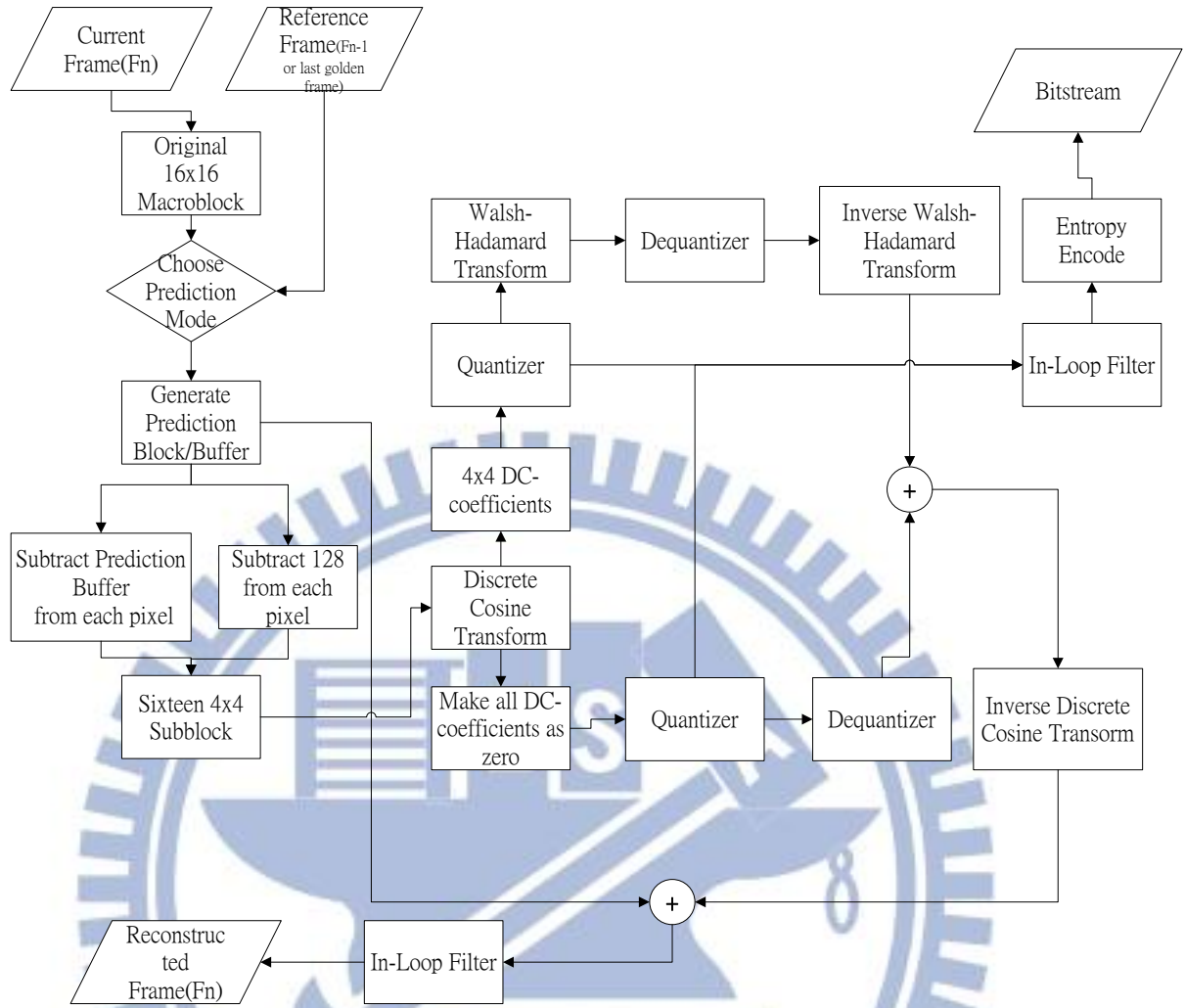


Figure 2.1 Flow diagram of WebM encoding process.

2.5.3 Process of Decoding

The decoder receives a compressed bitstream. First, the frame header (the beginning of the first data partition) is decoded. Then, the macroblock data occur in raster-scan order. These data come in two more parts. The first part is a prediction mode coming in the remainder of the first data partition. The other part comprises the data partition(s) for the DCT/WHT coefficients of the residue signal. Figure 2.3

shows the top-level hierarchy of the WebM video bitstream. For each macroblock, the prediction data must be processed before the residue. Each macroblock is predicted using one (and only one) of four possible frames, namely, the current frame, the immediately previous reconstructed frame, the most recent golden reference frame, and the recent alternate reference frame.

Regardless of the prediction method, the residue DCT signal is decoded, dequantized, reverse-transformed, and added to the prediction buffer to produce the reconstructed value of the macroblock, which is stored in the correct position of the current frame buffer. After all the macroblocks have been generated (predicted and corrected with the DCT/WHT residue), a filtering step is applied to the entire frame. The purpose of the loop filter is to reduce blocking artifacts at the boundaries between macroblocks and between subblocks of the macroblocks. Figure 2.3 shows the flow diagram of the WebM decoding process.

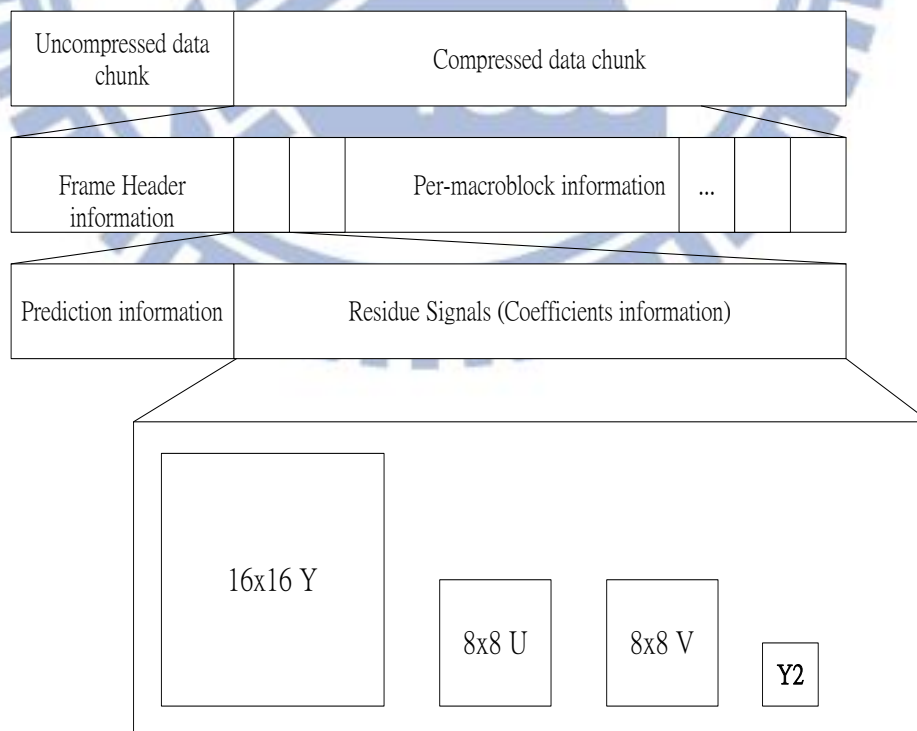


Figure 2.2 Top-level hierarchy of WebM video bitstream.

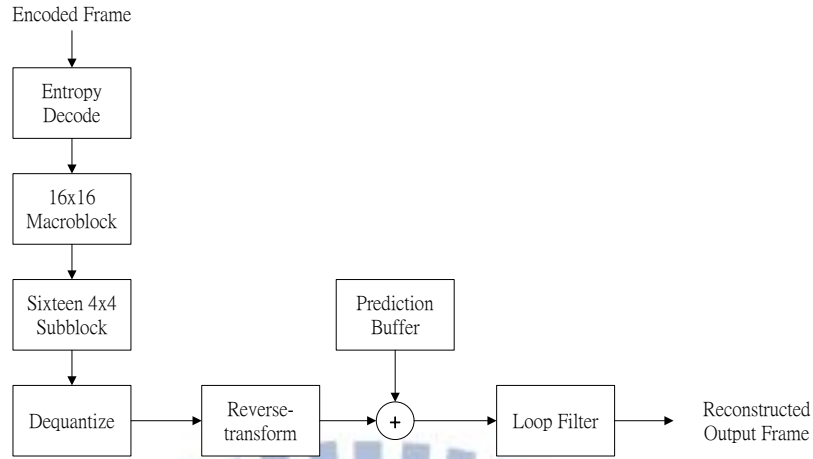


Figure 2.3 Flow diagram of WebM decoding process.

2.5.4 Region of Interest maps

The use of region of interest (ROI) maps is a way for applications to assign each macroblock in a frame to a region in WebM videos, and then set custom parameters such as quantization levels and filtering parameters. The VP8 video codec uses segment based adjustments to support changing the quantizer level and the loop filter level for a macroblock. It supports totally four different maps for each frame, so there could have up to four different maps in each frame. Macroblocks have its own map index, and these indexes also encode to be bitstreams by the tree coding. Figure 2.4 shows an example of ROI maps, where each block is a unit of map. Different colors mean different maps in this frame.

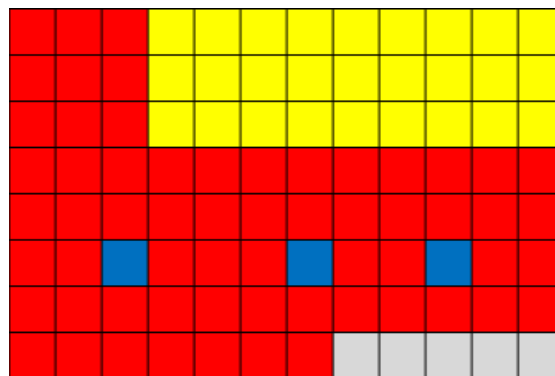


Figure 2.4 an Example of ROI maps of a frame.

2.5.5 Reference Frames

The VP8 video codec uses three types of reference frames for inter prediction: the *prior frame*, a *golden reference frame*, and an *alternate reference frame*. Overall, this design has a much smaller memory footprint on both the encoder and the decoder than designs with many more reference frames. More details of the golden reference frame and the alternate reference frame are illustrated below,

(A) Golden Reference Frame —

The VP8 video codec was designed to use one reference frame buffer to store a video frame from an arbitrary point in the past. This buffer is known as the *golden reference frame*. The VP8 encoder could use the golden reference frame in many ways to improve coding efficiency. One situation is that it can be used to maintain a copy of the background image when there are objects moving in the foreground part; by using the golden reference frame, the foreground part can be easily and cheaply reconstructed when a foreground object moves away. Another example is using the golden reference frame to encode back and forth cut of two scenes, where the golden reference frame buffer can be used to maintain a copy of the second scene. Finally, the golden reference frame can also be used for error recovery in a real-time video conference, or even in a multi-party video conference for scalability. Figure 2.5 shows an example of using the golden reference frame. In Figure 2.5, Frame 0 is a key frame and also a golden reference frame. Frame 1 through Frame 4 build a predictor using the prior frame. Frame 5 uses only Frame 0 as a reference. If any frames between Frame 1 to Frame 4 are lost, the VP8 video codec still can decode Frame 7 because it references only to Frame 0.

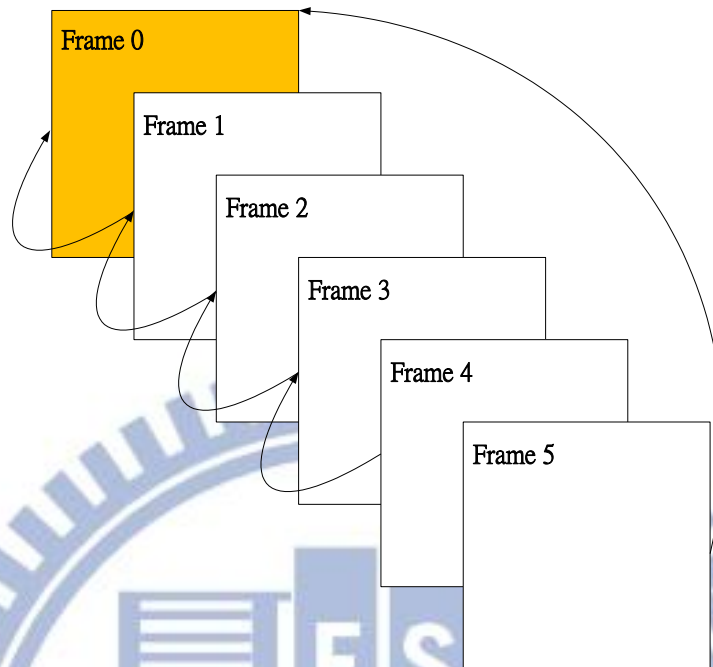


Figure 2.5 An example of the use of the golden reference frame.

(B) Alternate Reference Frame —

The VP8 alternate reference frame has much difference than other types of reference frames used in video compression. While reference frames usually are displayed to the user by the decoder, the VP8 alternate reference frame is decoded normally but may or may not be shown in the decoder. Because the alternate reference frames have an option of not being displayed, the VP8 encoder can use them to transmit any data that are helpful to compression. The flexibility in the VP8 specification allows many types of usage of the alternate reference frame for improving coding efficiency. For example, the VP8 video codec has a lack of B frames, which led to discussions in the research community about the ability to achieve high compression efficiency in the VP8 video codec. So, the VP8 video codec intelligently uses the golden reference frame and the alternate reference frames together to compensate for this problem.

2.5.6 VP8 Intra Prediction and Inter Prediction

To encode a video frame, a block-based video codec, such as the VP8 video codec, at first decomposes the frame into smaller segments called macroblocks. For each macroblock in the VP8 video codec, the encoder will predict redundant motion and color information based on previously processed macroblocks. The redundant information can be subtracted and transformed from the macroblock, resulting in more efficient compression. The VP8 encoder uses two prediction types: *intra prediction* and *inter prediction*. The intra prediction uses data within an encoded macroblock in this frame so it does not reference any previously encoded frames; and the inter prediction uses data from previously encoded frames, so the residual signal data are encoded using other techniques, such as transform coding.

(A) VP8 Intra Prediction Modes —

The VP8 video codec uses three types of macroblocks in intra prediction modes, 4×4 luma, 16×16 luma, and 8×8 chroma. Five intra prediction modes are shared by these macroblocks. The first is the H_PRED (horizontal prediction), which fills each column of the block with a copy of the left column. The second is the V_PRED (vertical prediction), which fills each row of the block with a copy of the row above. The third is the DC_PRED (DC prediction), which fills the block with a single value using the average of the pixels in the row above, A , and the column to the left, L (see Fig. 2.6). The fourth is the B_PRED, which divides a macroblock into sixteen blocks with each block having its own prediction modes. The last is the TM_PRED (TrueMotion prediction), which is a new compression prediction technique developed by On2 Technologies. We illustrate more details about TrueMotion prediction below.

In addition to the row A and the column L , TreMotion prediction uses the pixel C above and to the left of the block. Horizontal differences between pixels in A (starting from C) are propagated using the pixels from L to start each row. As mentioned above, the TM_PRED mode is unique to the VP8 video codec. Figure 2.6 uses an example 4×4 block of pixels to illustrate how the TM_PRED mode works, where C , A_x and L_x ($x = 0, 1, 2, 3$) represent reconstructed pixel values from previously encoded blocks, and X_{00} through X_{33} represent predicted values for the current block. The TM_PRED mode uses the following equation to calculate X_{ij} :

$$X_{ij} = L_i + A_j - C \quad (i, j = 0, 1, 2, 3).$$

C	A_0	A_1	A_2	A_3
L_0	X_{00}	X_{01}	X_{02}	X_{03}
L_1	X_{10}	X_{11}	X_{12}	X_{13}
L_2	X_{20}	X_{21}	X_{22}	X_{23}
L_3	X_{30}	X_{31}	X_{32}	X_{33}

Figure 2.6 An example of 4×4 block of pixels.

Although the above example uses a 4×4 block, the TM_PRED mode for 8×8 and 16×16 blocks works in the same way. The TM_PRED prediction mode is one of the more frequently used intra prediction modes in the VP8 video codec. Generally speaking, together with other intra prediction modes, the TM_PRED prediction mode helps the VP8 video codec to achieve very good compression efficiency, especially for key frames, which can only use intra modes.

(B) VP8 Inter Prediction Modes —

In the VP8 video codec, inter prediction modes are used only on inter frames (non-key frames). For any VP8 inter frame, typically three previously coded reference frames can be used for prediction. A typical prediction block is constructed using a motion vector to copy a block from one of the three frames. The motion vector points to the location of a pixel block to be copied. In most video compression schemes, a good portion of the bits are spent on encoding motion vectors; the portion can be especially large for videos encoded at lower data rates. The VP8 video codec encodes motion vectors very efficiently by reusing motion vectors from neighboring macroblocks. The VP8 video code uses a similar strategy in the overall design of inter prediction modes. For example, the prediction modes "NEARESTMV" and "NEARMV" make use of the last and second-to-last, non-zero motion vectors from neighboring macroblocks. And the prediction mode "ZEROMV" whose motion vectors in this macroblock is zero. These inter prediction modes can be used in combination with any of the three different reference frames.

In addition, the VP8 video codec has a very complicated, flexible inter prediction mode called SPLITMV. It is also a unique new compression prediction technique developed by On2 Technologies. This prediction mode was designed to enable flexible partitioning of a macroblock into sub-blocks to achieve better inter prediction. The SPLITMV prediction mode is very useful when objects within a macroblock have different motion characteristics. Within a macroblock encoded by the SPLITMV prediction mode, each sub-block can have its own motion vector. Similar to the strategy of reusing motion vectors at the macroblock level, a sub-block can also use motion vectors from neighboring sub-blocks above or left to the current block. This strategy is very flexible and can effectively encode any shape of sub-macroblock partitioning, and very efficiently. Figure 2.7 and Figure 2.8 illustrate an example of a macroblock using the SPLITMV prediction mode. In Figure 2.7, *NEW* represents a 4x4

block encoded with a new motion vector, and *LEFT* and *ABOVE* represent a 4×4 block encoded using the motion vector from the left and above, respectively. As can be seen from Figure 2.8, macroblocks have three different colors; and each color represents a segment with different motion vectors, so there exist three different motions in these macroblock.

<i>NEW</i>	<i>LEFT</i>	<i>LEFT</i>	<i>NEW</i>
<i>ABOVE</i>	<i>LEFT</i>	<i>LEFT</i>	<i>ABOVE</i>
<i>ABOVE</i>	<i>NEW</i>	<i>LEFT</i>	<i>ABOVE</i>
<i>ABOVE</i>	<i>ABOVE</i>	<i>LEFT</i>	<i>LEFT</i>

Figure 2.7 An example of the SPLITMV prediction mode.

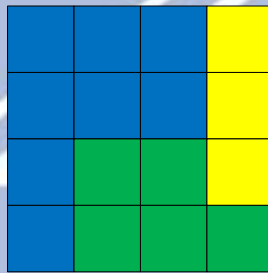


Figure 2.8 An example of the SPLITMV prediction mode.

Chapter 3

Data Hiding in WebM Videos for Covert Communication by Frequency Coefficient Modifications

3.1 Introduction

Due to the growth of computer network and audio/video compression technologies, many applications of digital media have emerged on the network. The preservation and transmission of secret information are interesting research topics. To solve such covert communication problems, the use of data hiding techniques is a good solution. In this way, we can hide secret data into cover media, and the hidden information is desirably imperceptible in general. Videos are suitable for use as cover media for this purpose because more data can be hidden in videos than in images or in other documents. In addition, because of the efficiency and good quality of the WebM video, some popular video sharing web sites, like YouTube, have already used WebM videos widely for user communications. Considering this popularity of the WebM video, we propose a data hiding method via WebM videos for covert communication in this study, which we describe in this chapter.

In Section 3.1.1, some relevant definitions are given, and in Section 3.1.2 the basic ideas of the proposed method are presented. In Section 3.2, the proposed data hiding method is described in detail, and the corresponding data extraction method is presented in Section 3.3. In Section 3.4, some experimental results are shown to prove

the feasibility of the proposed method. Finally, discussions and a summary of the proposed method are made in the last section of this chapter.

3.1.1 Problem Definition

When data hiding techniques via videos are applied for covert communication, the amount and imperceptibility of the hidden data are two major concerns. Furthermore, with the popularity of web applications, people give more and more attention to low bit rate videos. Therefore, an additional problem is how to hide data into videos in an optimal way to reduce the increase on the bit rate of the *stego-video*. Finally, the enhancement of the hidden secret security should also be taken into considerations.

3.1.2 Proposed Ideas

In the method proposed in this study for hiding data via WebM videos for covert communication, because the transform coding scheme in the VP8 video codec always conducts compression at the 4×4 resolution, we try to modify the WebM's frequency coefficients of the chroma color space in the compression result and generate data patterns for data hiding. In addition, the PSNR values are computed and compared with a threshold to optimize these changes for maintaining the video quality and the bit rate. For secret security enhancement, first we calculate the total number of macroblocks which can be used to embed data and the total size of the secret message. Then, we use a key together with a random number generator to select randomly data hiding positions in images, preventing a malicious user from figuring out the locations where the secret data are embedded.

There are two frame types in WebM videos, namely, *key frame* (I frame) and *prediction frame* (P frame). The data hiding technique we propose in this study

utilizes the prediction frame.

3.2 Embedding of Secret Data into WebM Videos

In this section, the proposed method for embedding secret data into the frequency coefficients of the WebM video will be described in detail. An illustration of the embedding process is shown in Figure 3.1. In Section 3.2.1, the idea of the proposed data embedding scheme is given, and in Section 3.2.2 the details of the corresponding process is described.

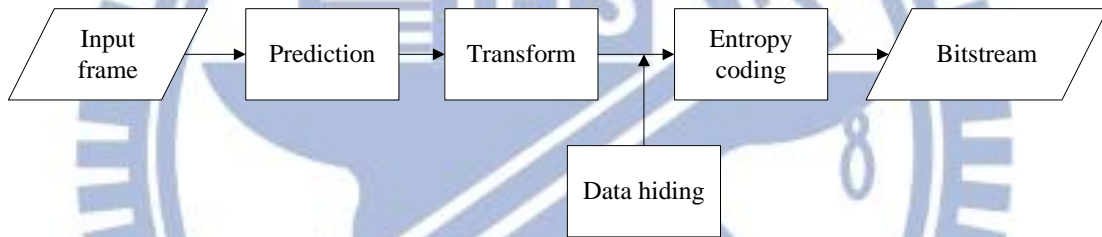


Figure 3.1 Illustration of the proposed data hiding method.

3.2.1 Idea of Proposed Method

Two main features of the proposed method are region-of-interest (ROI) map and frequency-coefficient pattern, whose functions for use in this study are described first below.

(A) *Region-of-Interest Map*

As mentioned in Section 2.5.4, the VP8 video codec supports up to four maps for each frame. Each macroblock has its own map index, and such an index is also encoded into the bitstream by tree coding. Here, we propose a scheme for assigning a map index for use as a *data extraction mark* to label macroblocks whose coefficients

are modified for embedding secret information. As a result, the proposed scheme can be used to indicate the macroblock positions in images where the secret information exists.

(B) Frequency Coefficient Patterns

As mentioned in Section 2.5.1, the macroblock-level data in a compressed frame in a WebM video is processed in a raster-scan order, and the macroblock is a square array of pixels whose Y components are 16×16 and U and V components are 8×8 . Each macroblock is decomposed further into 4×4 subblocks, so that every macroblock has sixteen Y subblocks, four U subblocks, and four V subblocks. Figure 3.2 shows one of the subblocks, whose size is 4×4 . The DCT (discrete cosine transform) and WHT (Walsh-Hadamard transform) are always performed to conduct compression at the 4×4 resolution in the VP8 video codec. And the pixel values in a subblock, after the DCT is conducted, will be transformed into frequency-domain coefficients, and the energy of the coefficient signals is “clumped” at the left-upper corner of the subblock.

In addition, after the quantization step with an adaptive quantization level is conducted, non-zero or zero coefficients will appear in the middle area of a quantized subblock. At this area of non-zero and zero coefficients, pre-defined data patterns may be generated automatically to replace them for imperceptible data hiding. Figure 3.3 shows an example of a subblock after performing the DCT and quantization. Furthermore, by the research results of the color theory [18], we know that human eyes have lower sensitivity on high-frequency signals and chrominance than on low-frequency signals and luminance.

According to the above discussions, we propose a data hiding scheme based on the DCT at the 4×4 resolution in this study, which modifies up to four coefficients on

the “positive-sloped diagonal line” of the 4×4 subblock of the quantized frequency coefficients using sixteen pre-defined 4×4 patterns to represent the message information to be embedded. Here, by the *positive-sloped diagonal line*, we mean those yellow-colored squares in the 4×4 coefficient matrix (corresponding to a subblock) shown in Fig. 3.2 or those red-colored ones shown in Fig. 3.3.

There are two reasons why we do not choose the coefficients from the upper left nor from the lower right part of the coefficient matrix to conduct pattern replacement for data embedding there. First, if the quantization level used for quantizing the coefficients in the lower right portions is too large, modifications of the coefficients there will cause too much distortion in the resulting image, allowing one to perceive any modification that has been done, so that imperceptibility would not be achieved. Second, the coefficients in the upper left portion yielded by the DCT and the quantization process are usually non-zero values; therefore, it is almost impossible for the message data to match the pre-defined patterns well without modifying the coefficients.

Considering the capacity of hiding data and the above reasons, the proposed method uses the positive-sloped diagonal lines of all the subblocks of the chroma color channel for data embedding.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure 3.2 An example of subblocks with yellow coefficients composing a positive-sloped diagonal line.

81	20	6	-2
11	4	0	0
1	0	0	0
0	0	0	0

Figure 3.3 An example of a subblock after performed DCT and quantization with red coefficients composing a positive-sloped diagonal line.

3.2.2 Process for Embedding Secret Data

In this section, we will describe the detailed algorithm of the proposed method for hiding secret message data into cover videos by changing the frequency coefficients into pre-defined patterns. A flowchart of the proposed data embedding process is shown in Figure 3.5.

Beforehand, we define in the following the aforementioned 16 data patterns for use in the proposed algorithms where we use the notations N and 0 to denote the meanings “non-zero” and “zero,” respectively.

Data pattern i ($i = 0$ to 15): a 4×4 block with its positive-sloped diagonal line being filled with four symbols $S_4S_3S_2S_1$ of N’s and 0’s, which correspond to the binary value $b_4b_3b_2b_1$ of i in the following way:

$$\text{if } b_j = 0, \text{ then } S_j = N; \text{ otherwise, } S_j = 0, j = 1, 2, 3, 4.$$

Figure 3.4 illustrates the 16 data patterns. For example, when $i = 3$, the corresponding binary value is $i = 3_{10} = 0011_2$, so we define *pattern 3* as the 4×4 block with its positive-sloped diagonal line being filled with the four symbols $S_4S_3S_2S_1 = 00NN$. And when $i = 10$, the corresponding binary value is $i = 10_{10} = 1010_2$, so we define *pattern 10* as the 4×4 block with its positive-sloped diagonal line being filled with $S_4S_3S_2S_1 = NON0$.

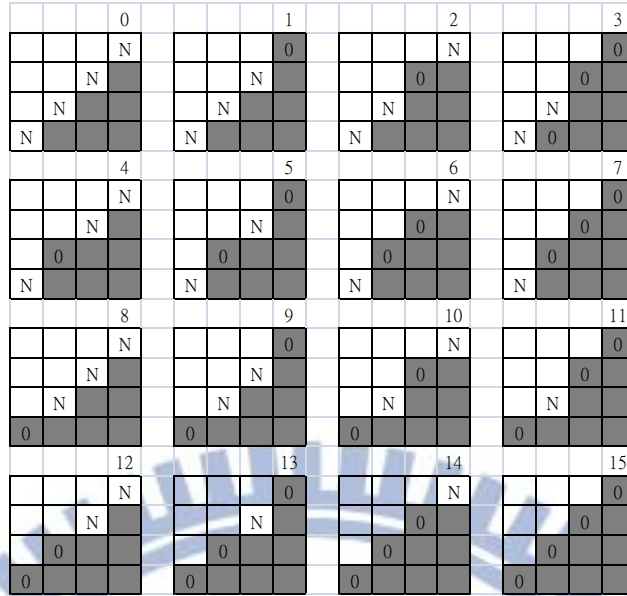


Figure 3.4 The sixteen data patterns for use to embed message data.

Algorithm 3.1 Process for computing the data hiding capacity of a video sequence.

Input: a video sequence V and a pre-selected threshold T .

Output: the number C of macroblocks in V which may be used to embed data patterns without causing intolerable distortion.

Steps.

1. Perform the following steps for each macroblock MB in the prediction frames of V .
 - 1.1 Save the original quantized coefficients of the chroma color channels (including the U channel and the V channel) in the macroblock MB .
 - 1.2 Check the coefficients of each subblock SB of the chroma color channels:
 - if the original coefficients of SB do not satisfy data pattern 0, then modify them to be so by changing 0 in them to be 1;
 - else, do nothing.
 - 1.3 (*Computing the resulting distortion*) Calculate the mean square quantization error (MSQE) between the saved content MB_o of the original macroblock

MB and the content MB_o' of the modified macroblock MB' of the chroma color channels:

$$MSQE_i = \sqrt{(MB_o' - MB_o)^2} \quad (3.1)$$

where $i = 1$ and 2 represent the U channel and the V channel, respectively, and each of MB_o and MB_o' means an 8×8 vector of coefficients.

- 1.4 Calculate the following value of the peak signal-to-noise ratio (PSNR) $PSNR_i$ of the chroma color channels for $i = 1$ and 2 :

$$PSNR_i = 10 \times \log \left(\frac{S_{peak}^2}{MSQE_i} \right) \quad (3.2)$$

where S_{peak} means the maximum possible pixel value of the image.

- 1.5 Calculate the average PSNR value $PSNR_{avg}$ of the chroma color channels:

$$PSNR_{avg} = \frac{\sum_{i=1}^2 PSNR_i}{2}. \quad (3.3)$$

- 1.6 (*Checking the data embeddability of the macroblock*) Use the ROI map index to label the modified macroblock MB' in the following way:

if $PSNR_{avg}$ is smaller than the pre-selected threshold T , then set the ROI map index value to be 1, meaning that macroblock MB is *data-embeddable*;

else, use the default value of the ROI map index which is 0, meaning that macroblock MB is *non-data-embeddable*.

2. Increment the value C by one if the ROI map index is set to be 1.
3. Repeat Steps 1 and 2 until all macroblocks are processed.

In Step 2 above, if the ROI map index is set to be 1, it means that this macroblock can be used to embed data without causing intolerable distortion in the resulting macroblock. Also, the number C is used to specify the data hiding capacity

of this video sequence in unit of macroblock. In addition, the Parseval theorem [19] states that mean square error (MSE) in the pixel domain is equivalent to the mean square quantization error (MSQE) in the DCT domain because the DCT is a normalized orthogonal transformation. So in Steps 1.3 and 1.4 above, we may also use the original PSNR definition described by Eq. (3.3) below to calculate the PSNR values:

$$PSNR = 10 \times \log \left(\frac{S_{peak}^2}{MSE} \right). \quad (3.4)$$

where S_{peak} means the maximum possible pixel value of the image.

With the data embedding capacity C computed, we can now describe the proposed method for data embedding as an algorithm in the following.

Algorithm 3.2 *Process for embedding secret data into a WebM video.*

Input: a video V , a secret key K , a random number generator f , and a secret message S .

Output: a stego-video V' .

Steps.

Stage 1 --- initialization.

1. Process V to compute the data hiding capacity C in V by performing Algorithm 3.1.
2. (*Randomizing the secret message*) Transform the secret message S into a binary string B , use the secret key K as a seed to generate a sequence Q of random numbers using the random number generator f , and randomize B with Q to get a randomized binary string B' .
3. Calculate the total number N of macroblocks which are needed for embedding B'

by:

$$N = \frac{\text{the length of } B'}{32}. \quad (3.5)$$

4. Use the secret key K and the random number generator f to generate a sequence RS of N random integer numbers with C as the maximum number in the sequence, and sort them into an ascending order.
5. Divide B' into a linear array A of 4-bit segments.

Stage 2 --- embedding message data into the video.

6. Perform the following steps to embed message S into each *unprocessed* macroblock MB in every prediction frame of V , assuming V is large enough to embed the entire message.

6.1 Save the original quantized coefficients of the chroma color channels in MB .

6.2 Check the coefficients of each subblock SB of the chroma color channels:
if the original coefficients of SB do not satisfy data pattern 0, then modify them to be so by changing 0 in them to be 1;
else, do nothing.

6.3 Calculate the mean square quantization error (MSQE) between the saved content MB_o of the original macroblock MB and the content MB_o' of the modified macroblock MB' of the chroma color channels:

$$MSQE_i = \sqrt{(MB_o' - MB_o)^2} \quad (3.6)$$

where $i = 1$ and 2 represent the U channel and the V channel, respectively, and each of MB_o and MB_o' means an 8×8 vector of coefficients.

6.4 Calculate the value of the peak signal-to-noise ratio (PSNR) $PSNR_i$ of the chroma color channels for $i = 1$ and 2 :

$$PSNR_i = 10 \times \log \left(\frac{S_{peak}^2}{MSQE_i} \right) \quad (3.7)$$

where S_{peak} means the maximum possible pixel value of the image.

6.5 Calculate the average PSNR value $PSNR_{avg}$ of the chroma color channels:

$$PSNR_{avg} = \frac{\sum_{i=1}^2 PSNR_i}{2}. \quad (3.8)$$

6.6 (*Checking the data embeddability of the macroblock*) Use the ROI map index to label the modified macroblock MB' in the following way:

if $PSNR_{avg}$ is smaller than the pre-selected threshold T , then set the the ROI map index value to be 1, meaning that macroblock MB is *data-embeddable*;

else, use the default value of the ROI map index which is 0, meaning that macroblock MB is *non-data-embeddable*.

6.7 Increment the value of a pre-defined *random selection counter* C_v by one if the ROI map index of this macroblock is set to be 1.

6.8 (*Embedding the secret data patterns at random locations in the input video*)

If C_v is equal to the next unprocessed random number in RS (meaning that the currently macroblock is chosen randomly for message hiding), then conduct the following steps to embed eight 4-bit segments (=32 bits) of the message data; if not, go to Step 6.9.

(1) Take eight unprocessed 4-bit elements from A , denoted as A_1 through A_8 , and for each element A_i , define for it a corresponding data pattern P_i such that if $A_i = (b_1b_2b_3b_4)_2 = d_{10}$, then P_i is just data pattern d (e.g., if $A_i = 1001_2 = 9_{10}$, then $P_i =$ data pattern 9).

(2) Check the coefficients of each of the eight corresponding subblocks of the chroma color channels with four in the U channel and the other four

in the V channel, denoted as $SB_j, j = 1, 2, \dots, 8$:

if the original coefficients of SB_j do not match those of data pattern P_i , then modify them to be so by changing those mismatching ones in SB_j to be the corresponding ones of data pattern P_i ;
else, do nothing.

6.9 Embed an *ending pattern* in the next macroblock by setting its ROI map index to be 1 if the entire secret message in array A has been embedded (i.e., if all elements in A have been processed); otherwise, go to Step 6 to repeat the above process.

The random sequence RS generated in Step 4 above is used to represent positions where the secret message can be embedded. In the same step, the value N represents the number of macroblocks we need to embed the entire secret message. Selecting N elements from RS and using them in Step 6.8 means that the message data are embedded into random positions. By this way, we can reduce the opportunity for an attacker to get the secret message. Also, because the VP8 encoder always encodes frames in raster-scan order, we sort the N elements in an ascending order.

In Step 5, the proposed hiding method is based on the use of the 16 pre-defined data patterns, so we have to divide the randomized binary string B' into 4-bit segments to correspond to the 4-bit message data embeddable in the positive-sloped diagonal line of each data pattern. For example, if a string is 000111000101110000111000001110101111, after we divide the string into a data pattern array, the resulting elements in the data pattern array A are 1, 12, 5, 12, 3, 8, 3, 10, and 15. In Step 6.7, the initial value of the random selection counter is set to be 0. In Step 6.9, the ending pattern is used to specify when to stop extracting message data in the extraction process.

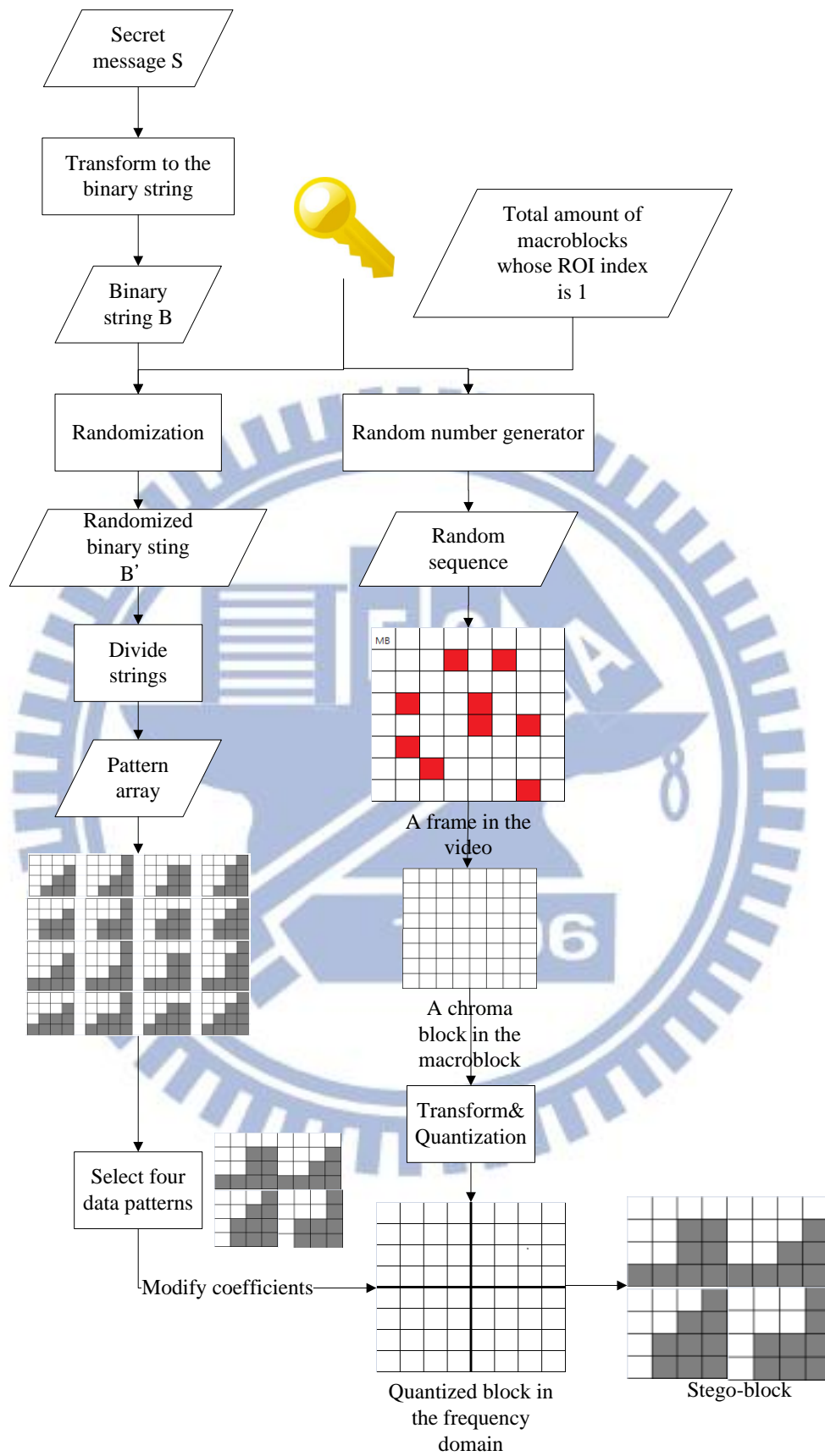


Figure 3.5 A flowchart of embedding process.

3.3 Extraction of Secret Data from WebM Videos

In this section, the proposed method for extracting the hidden message data from a stego-video of the WebM format is described. A secret key, which presumably was used in Algorithm 3.2, is used again here as a seed of a random number generation. Extraction of the information hidden in the frequency domain is performed based on the result of the random number generation. More details are described in the following as an algorithm — Algorithm 3.3. An illustration of the algorithm is included in Figure 3.6.

Algorithm 3.3 *Process for extracting secret message data from a stego-video.*

Input: a stego-video V ; and the secret key K and the random number generator f used in Algorithm 3.2.

Output: a secret message S .

Steps.

Stage 1 --- initialization.

1. Define the number C as the data embedding capacity of V , check each macroblock in V , and increment C by one if the ROI map index of this macroblock set to be 1.
2. Use the secret key K and the random number generator f to generate a sequence RS of random integer numbers with C as the maximum number in the sequence.

Stage 2 --- extracting message data from the video.

3. Check the macroblocks in every prediction frame of the stego-video V , increment a pre-defined *random selection counter* C_v by one if the ROI map index of the currently-processed macroblock has been set to be 1 and check the value of C_v :

if C_v of the current processing macroblock is equal to the next unprocessed random number in RS (meaning that message bits has been embedded in the currently-processed macroblock), then perform the following steps to extract eight 4-bit segments of the message data; otherwise repeat Step 3.

3.1 Check the coefficients of each of the eight corresponding subblocks of the chroma color channels (four in the U channel and the other four in the V channel), denoted as $SB_j, j = 1, 2, \dots, 8$: if the coefficients of SB_j match those of data pattern i , then record the code of *data pattern i* as an *extracted result R_i* (e.g., if data pattern $i = 3$ is matched by SB_j , then $R_i = 3_{10} = 0011_2$).

3.2 Append R_i to an *extracted data pattern D* , which is set empty initially.

3.3 Save D into an array A , and repeat Step 3 if D is different from the ending pattern; otherwise, continue.

4. Sort the array A in an ascending order.
5. Transform all elements of array A and concatenate them to form a binary string S_b .
6. (*Reorganizing the secret message*) Use the secret key K as a seed to generate a sequence Q of random numbers using the random number generator f , de-randomize S_b with Q to get another binary string S_b' , and transform S_b' into a character form for use as the desired output secret message S .

3.4 Experimental Results

In our experiments, the proposed data hiding algorithm utilizes the VP8 video codec of version 0.9.7 for video compression by the DCT. Some configuration parameters of the VP8 video codec used in the compression process are shown in

Table 3.1. Several video clips, including Bus, Container, Silent, Tempete, and Waterfall with the CIF (352×288 pixels) format, are used in our experiments. The secret message with the size of 468 bytes used in the experiments is shown in Figure 3.7.

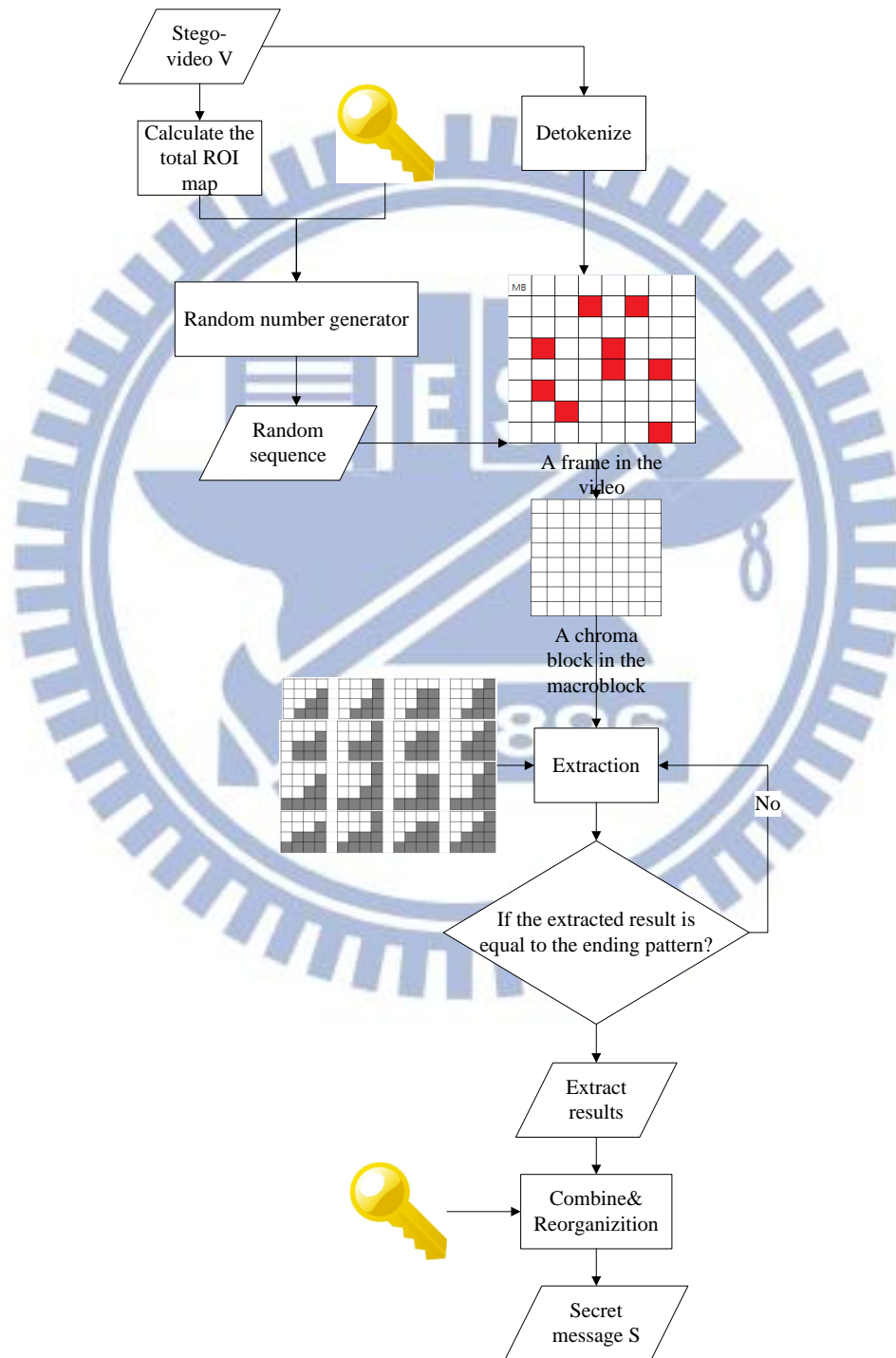
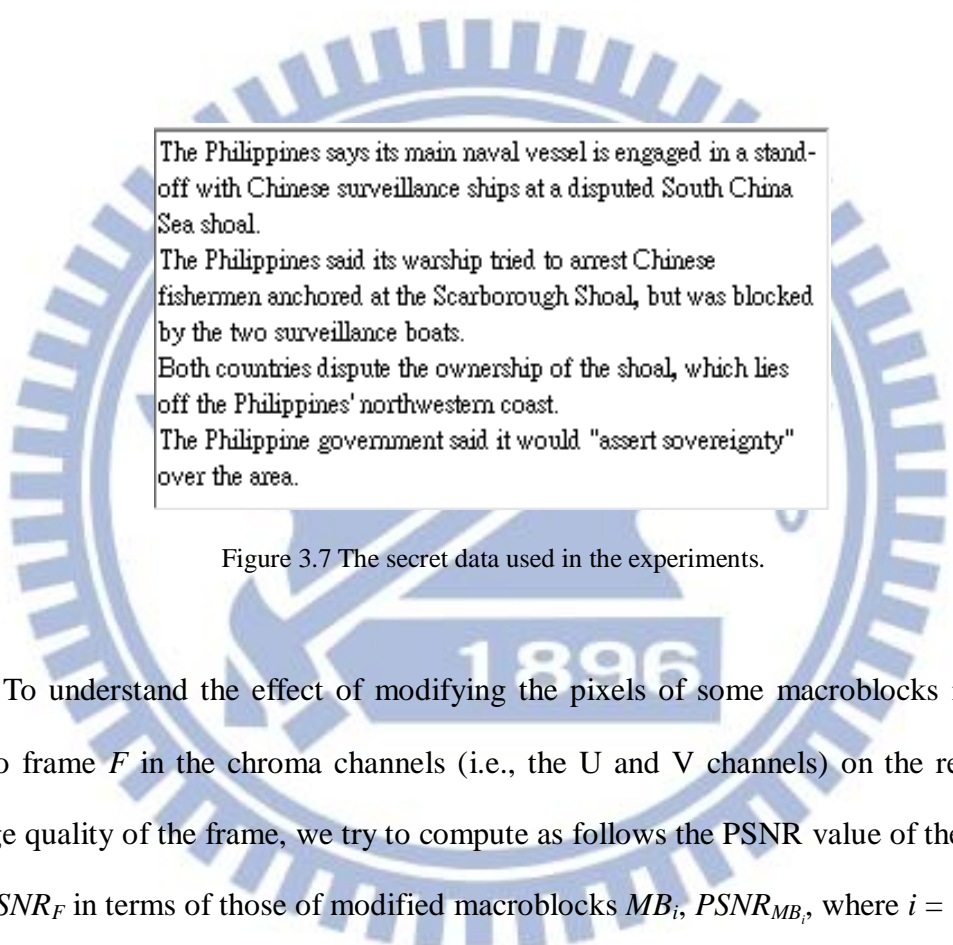


Figure 3.6 A flowchart of proposed message data extraction process.

Table 3.1 Configuration parameters used in this study.

<i>Profile</i>	Baseline
<i>Number of frames to be coded</i>	50
<i>Encode Quality</i>	-- good
<i>Speed</i>	-- cpu-used = 4
<i>Rate Control</i>	-- max-q = 20



The Philippines says its main naval vessel is engaged in a stand-off with Chinese surveillance ships at a disputed South China Sea shoal.
 The Philippines said its warship tried to arrest Chinese fishermen anchored at the Scarborough Shoal, but was blocked by the two surveillance boats.
 Both countries dispute the ownership of the shoal, which lies off the Philippines' northwestern coast.
 The Philippine government said it would "assert sovereignty" over the area.

Figure 3.7 The secret data used in the experiments.

To understand the effect of modifying the pixels of some macroblocks in each video frame F in the chroma channels (i.e., the U and V channels) on the resulting image quality of the frame, we try to compute as follows the PSNR value of the frame F , $PSNR_F$ in terms of those of modified macroblocks MB_i , $PSNR_{MB_i}$, where $i = 1, 2, \dots, n$ with n being the total number of modified macroblocks in frame F [20]:

$$PSNR_F = -10 \times \log \left[\frac{256}{FS} \sum_{i=1}^n \frac{1}{10^{\frac{PSNR_{MB_i}}{10}}} \right] \quad (3.9)$$

where FS specifies the frame size in unit of pixel. Figure 3.8 shows the computation results for the above-mentioned video clips. In Figure 3.8, the PSNR values of some frames of the tested videos can be seen to be lower than others, such as the 8th, 9th,

12th, and 49th frames in the video Tempete, the 24th and 34th frames in the video Waterfall, the 14th frame in the video Bus, and the 27th frame in the video Container.

Some results of comparisons between the original frames and the stego-frames resulting from applying Algorithm 3.3 to the tested video clips are shown in Figure 3.9 through Figure 3.12. Although the PSNR values of some frames of the tested videos can be seen to be lower than others in Figure 3.8, they do not cause intolerable distortions as can be seen from the figures; therefore, imperceptibility is achieved. Finally, the extracted secret messages resulting from applying Algorithm 3.3 is shown in Figure 3.13, where a correct secret message is seen in Fig. 3.13(a) which was extracted with a right key, and an incorrect one is seen in Fig. 3.13(b) which was extracted with an erroneous key.

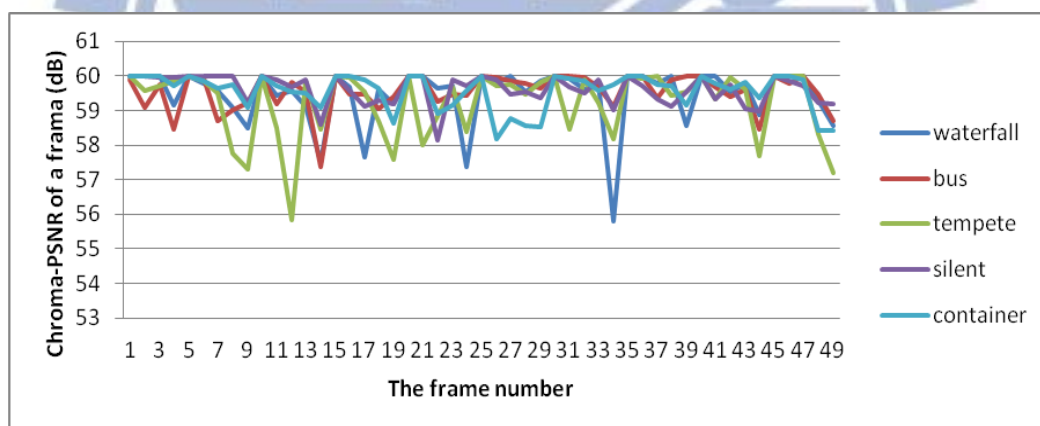


Figure 3.8 The experimental result of computing the PSNR values of chroma frames of tested videos.



Figure 3.9 The 8th, 9th, 12th, and 49th frames of original video (left) Tempete and stego-video (right).



Figure 3.9 The 8th, 9th, 12th, and 49th frames of original video (left) Tempete and stego-video (right).
(cont'd)

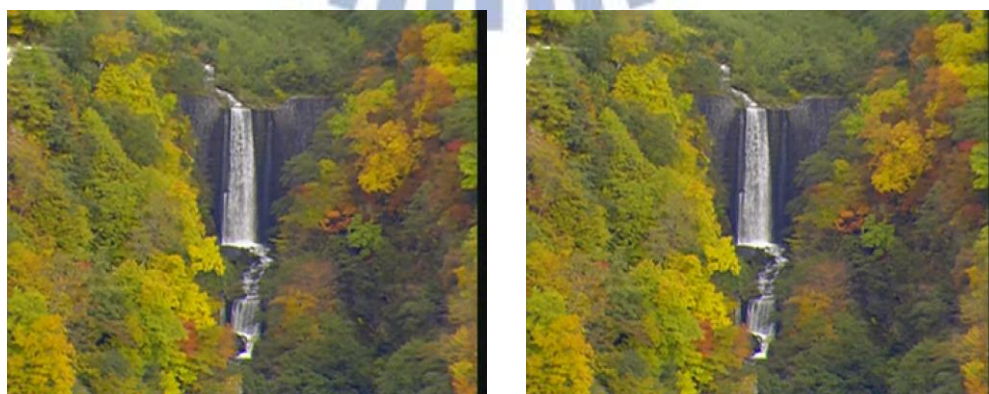


Figure 3.10 The 24th and 34th frames of original video Waterfall (left) and stego-video (right).



Figure 3.10 The 24th and 34th frames of original video Waterfall (left) and stego-video (right). (cont'd)



Figure 3.11 The 14th frame of original video Bus (left) and stego-video (right).



Figure 3.12 The 27th frame of original video Container (left) and stego-video (right).

The Philippines says its main naval vessel is engaged in a stand-off with Chinese surveillance ships at a disputed South China Sea shoal.

The Philippines said its warship tried to arrest Chinese fishermen anchored at the Scarborough Shoal, but was blocked by the two surveillance boats.

Both countries dispute the ownership of the shoal, which lies off the Philippines' northwestern coast.

The Philippine government said it would "assert sovereignty" over the area.

(a)

†>X8i° 22Êö&Wf¶p “Yβ+ri/ 3Øk’ x&uGÖ-
 VoW+÷×Uc · ÜÄGäuý&KÜðèocèÀøpèúQyòÚ&
 <^Üî-¶oós 0zÇ&^òVÖ&÷iuí?o ù~Az»
 þ'qw&Á×|~mýláxÖβðJš ‘dx- iÜBz&yÜüCβ&Súø°
 · WÊ™%pyíuYíñ@SèñD;DsySúOβÉ±>
 þRXYç@Éú^A7~§ mî~iy]5Üy9óó?pc~→>Vó†™
 íYî-ò×-B«ÜÊâ —èyúj>èqíÿ\$fuÿ#æ2Ü&Eý&ð- 8U-
 è×Ü-i*?Ä@-D;Ü^Üü?|@ñí|â-9-Ü×Tzn- É °&Mó
 «ö|™
 aó7kÿ™yA&wJŽ=zβ™ y” f’Kí “Èçyñ&Húú+{ù-¿É[t
 <èÇp[È#³¿ð ÷ÄzPKí):

(b)

Figure 3.13 Extracted secret messages. (a) The correct secret message with a right key. (b) The incorrect secret message with an erroneous key.

3.5 Discussions and Summary

The performance of the data hiding algorithm evaluated in terms of the *number of hidden bits* (NHB), the *peak-signal-to-ratio increase* in the chroma channels (PSNRI), and the *video size increase* (VSI) for several video sequences are shown in Table 3.2. Also shown in the table is the ratio of the NHB to the VSI, which indicates the efficiency of the data hiding scheme. From the table, it can be observed that the proposed method can embed the secret message into WebM videos imperceptibly with small video size increases. Also, the number of hidden bits is larger than the increased number of bits in the resulting enlarged video file, also indicating the good efficiency of the proposed data hiding method.

As a summary, we have proposed a data hiding method via WebM videos by frequency coefficient modifications. The data hiding method not only considers the

data hiding capacity and imperceptibility, but also the security issue. Therefore, the method is suitable for covert communication applications via WebM videos.

Table 3.2 Values of NHB, PSNRI, and VSI of several video clips.

	Waterfall	Bus	Tempete	Silent	Container
<i>Number of hidden bits (NHB)(bits)</i>	76896	52640	102592	5024	14112
<i>Average of PSNRI (dB) of frames</i>	-0.5467	-0.4545	-0.7705	-0.9837	-0.4339
<i>VSI (bits)</i>	27928	20245	34204	2876	5307
<i>Ratio of VSI/NHB</i>	0.3632	0.3846	0.3334	0.5725	0.3761



Chapter 4

Authentication of Surveillance Videos by Motion Object Analysis

4.1 Introduction

With the progress of video compression technologies and efficient video coding standards, digital videos nowadays have become more and more popular in recent years. Also, with the convenience of the Internet, many people transmit digital videos through the Internet. WebM videos are getting more and more popular and have been used in many applications, such as video surveillance, multimedia interchanges, etc.

For example, a video surveillance system is usually used to monitor indoor or outdoor environments to achieve the following goals: monitoring a gate and recording related information for events happening around, monitoring the traffic conditions on roads, querying a specific event in a surveillance video, and so on. However, digital videos may as well be acquired and tampered with by malicious users. And if the contents of stored videos have been tampered with, it may cause serious flaws, especially those of surveillance videos, which often include monitored private environments not to be accessed by un-authenticated users. Therefore, it is necessary to have a security mechanism to *authenticate* the integrity and fidelity of surveillance videos. In this chapter, a method for authentication of the contents of surveillance videos is proposed.

In Section 4.1.1, the problem definition is given. In Section 4.1.2, the idea of the proposed method is presented. In Section 4.2, the proposed process for generating

authentication signals is described. In Section 4.3, the proposed process for embedding authentication signals in surveillance videos is described. And in the Section 4.4, the proposed method for authentication of the contents of videos is described. Some experimental results are shown in Section 4.5. Finally, some discussions and a summary are given in the last section of this chapter.

4.1.1 Problem Definition

About the video authentication problem dealt with in this study, as mentioned, the content in a WebM surveillance video may be tampered with by malicious users, aiming at removing illegal behaviors. So, it is necessary to develop a method for the WebM surveillance video to verify whether the video content has been tampered with or not.

There are two main issues in this problem: 1) how to detect tampered regions in a WebM surveillance video automatically; and 2) how to generate a *protected surveillance video* with authentication signals embedded. The main goal of the proposed authentication system is not only to detect whether a protected surveillance video has been tampered with, but also to locate and mark the tampered regions in the video frames.

4.1.2 Proposed Ideas

In order to detect tempering in a digital video, *authentication signals* are generated for each video frame which contains *motion objects*. For this, the proposed video authentication system uses the *prediction modes* and *motion vectors* to detect motion objects in a surveillance video. The system then groups the motion objects into *motion regions* for use to generate authentication signals, which then are embedded into the surveillance video by modifying the frequency coefficients of the

macroblocks of the video. In this way, the system generates a *protected* surveillance video.

If a malicious user gets the protected surveillance video, he/she may have a chance to illegally modify the contents of the video. To authenticate such possible attacks, the authentication signals embedded in the protected surveillance video are extracted, using the proposed authentication method, to verify the integrity and fidelity of the motion objects, which might have been removed by malicious users, to avoid disputes in laws.

4.2 Generation of Authentication Signals by Motion Contents

In this section, the proposed method for generation of authentication signals is described. In Section 4.2.1, the principle of authentication signal generation is described at first, and in Section 4.2.2, the detail of the proposed method for generating authentication signals is then presented.

4.2.1 Principle of Authentication Signal Generation

In Section 2.5.6, we have reviewed the prediction modes used in the WebM video. In this study, we propose a method for detection motions in WebM videos by analyzing the prediction modes and motion vectors in each prediction frame. The basic idea is that macroblocks in a frame have different prediction modes and motion vectors for motion object parts and still object parts, respectively. So, prediction modes are suitable for use to generate authentication signals and verify motion objects.

The contents of motion regions usually change greatly both in the time domain

and in the spatial domain. In the time domain, movements in motion regions yield a lot of changes, which need to be described. In the spatial domain, motion regions contain motion objects which might be humans, cars, and so on; these motion objects might contain lots of details that need to be encoded.

To reduce the coding cost in the prediction step, changes in the spatial domain are generally encoded by the best prediction mode in the motion regions of the macroblocks. Therefore, the prediction mode of a motion macroblock is different from a still macroblock. Moreover, changes in the time domain make motion vectors of motion macroblocks larger because the contents of these macroblocks are quite different from each other. Figure 4.1 shows an example of different contents of the video using different prediction modes. As shown in the figure, because the motorcycle is a motion object, the prediction modes and motion vectors around it are different from those of the other areas in this video frame.



Figure 4.1 An example of different prediction modes in the content of a video.

Based on the clues mentioned above, we choose prediction modes and motion

vectors as features for detecting motion objects in a prediction frame, and then use a region growing algorithm to group them into motion regions in the surveillance video. Figure 4.2 shows an example of motion regions generated by the proposed method for motion detection.

Besides, after generating motion regions by these features, some noise blocks caused by variations of lights and shadows may be included in the detected motion regions. We call macroblocks containing such noise in the detected motion regions as *noise macroblocks*. Motion vectors of these noise macroblocks usually have smaller motion vectors. Therefore, we use motion vectors to eliminate such noise. An example of noise macroblocks is illustrated in Figure 4.3.



Figure 4.2 An example of motion regions.

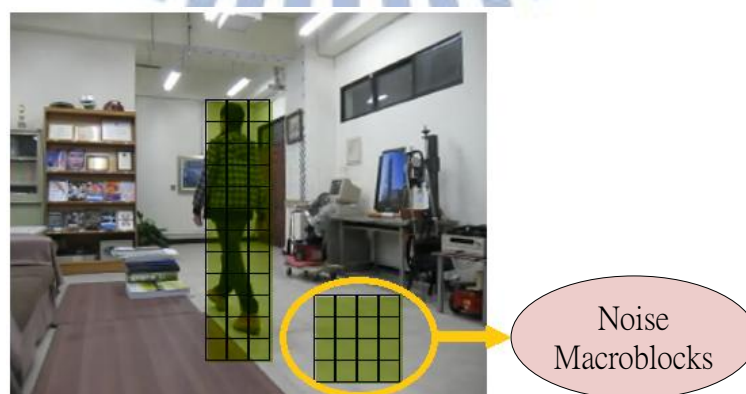


Figure 4.3 An example of noise macroblocks.

4.2.2 Process of Authentication Signal Generation

In this study, each frame is treated as a unit for authentication. If the prediction frame P of an input video is not still, it will have its own authentication signals which comprise two parts. The first part is the index of P used to indicate which frame of this authentication signal belongs to. And the second part is the motion region which is used to detect whether the contents of the video have been tampered with or not. We analyze the prediction modes in each motion region to generate a grade as the authentication signal for a key frame whose previous frame is a motion frame with motion regions. The detailed algorithms are described below.

Algorithm 4.1 Process for generating authentication signals of the prediction frame.

Input: a prediction frame F of a WebM video.

Output: authentication signals S_a for the frame.

Steps.

Stage 1 --- detecting motion objects in the prediction frame.

1. Perform the following steps for each macroblock MB in the prediction frame F .
 - 1.1 If the prediction mode of MB is *inter coded* such as the NEWMV, NEARMV, NEARESTMV, and SPLITMV modes, record MB as a *motion block*; else, record MB as a *still block*.
 - 1.2 If the prediction mode of MB is *intra coded* such as the B_PRED, H_PRED, DC_PRED, V_PRED, and TM_PRED modes, decide MB as a *motion block* or a *still block* according to the following steps.
 - 1.2.1 Name the eight neighboring macroblocks as A through H , as depicted in Figure 4.4.
 - 1.2.2 Define the *macroblock gain* G_i for each of A through H in the

following way.

- (1) For A , B , C , and D , if the prediction mode of this macroblock is the NEWMV or SPLITMV mode, then set the value of G_i to 10; if the prediction mode of this macroblock is the NEARMV or NEARESTMV mode, then set the value of G_i to 5; otherwise, to 0.
- (2) For E , F , G , and H , if the prediction mode of this macroblock is the NEWMV or SPLITMV mode, then set the value of G_i to 5; if the prediction mode of this macroblock is the NEARMV or NEARESTMV mode, then set the value of G_i to 2; otherwise, to 0.

1.2.3 Calculate a value named *score* according to the following equation:

$$score = \sum_{i=A}^H G_i . \quad (4.1)$$

1.2.4 If the value *score* is larger than a pre-defined threshold T , record MB as a *motion block*; otherwise, as a still block.

2. Repeat Step 1 until each macroblock has been processed.
3. Perform the following steps to group the motion blocks in F detected in the last two steps to get a set R of *candidate motion regions*.
 - 3.1 Apply a region growing algorithm to F while regarding each motion block in F as a *pixel* to get a set of connected components as the desired set R .
 - 3.2 For each connected component R_i' found above, perform the following two steps.
 - 3.2.1 Find the four extreme x -coordinates and y -coordinates of all the pixels in R_i' .
 - 3.2.2 Use the four extreme x - and y -coordinates to generate an upright

rectangular shape to circumscribe R_i' , and take the resulting rectangular shape as a candidate motion region R_i .

Stage 2 --- deciding the final motion regions in the prediction frame.

4. Perform the following steps for each candidate motion region R_i in R .
 - 4.1 Check the motion vectors of all the macroblocks for each edge e_j , where $j = 1, 2, 3, 4$ specifying the east, north, west, and south edges of R_i .

4.1.1 Calculate the total number N of macroblocks of the edge e_j .

4.1.2 Compute MV_i by the following formula:

$$MV_i = |mv_{ix}| + |mv_{iy}|, \quad (4.2)$$

where $i = 1$ to N and (mv_{ix}, mv_{iy}) denotes the motion vector of the i -th macroblock.

4.1.3 If MV_i is larger than a pre-defined threshold T' , then set the *motion score* M_i of MV_i to 1 ($i = 1$ to N); else, set M_i to 0.

4.1.4 Compute the average of motion scores, denoted by S , by the following formula:

$$S = \frac{\sum_{i=1}^N M_i}{N}; \quad (4.3)$$

and if S is smaller than 0.5, then *abandon* these macroblocks on the edge e_j to *shrink* R_i .

4.2 Repeat Step 4.1 until all the four edges are processed and none has been shrunk (meaning that all the four edges are kept) or until no macroblock remains in R_i (meaning that R_i is shrunk to disappear).

4.3 Save relevant information of the motion region R_i which is not shrunk any more — the index of the current frame, the index of the first region block, and the index of the last region block, into S_a .

5. Repeat Step 4 until all candidate motion regions in R are processed.

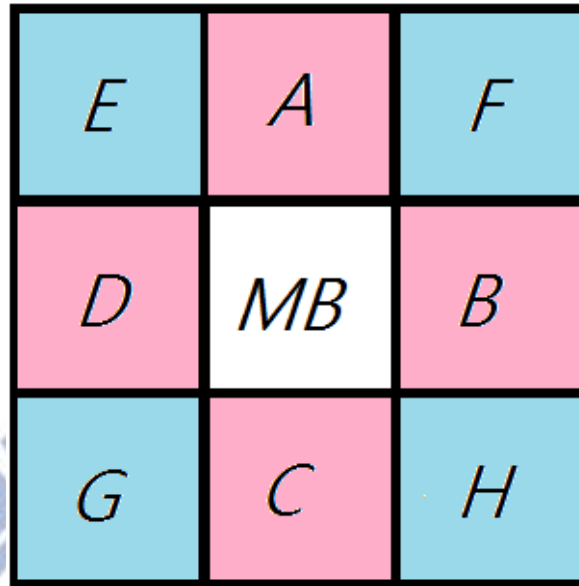


Figure 4.4 The notations of the eight neighboring macroblocks of MB .

Algorithm 4.2 Process for generating authentication signals of the key frame.

Input: a key frame F of the WebM video.

Output: authentication signals S_a .

Steps.

1. Get a set of the motion regions R from the last prediction frame, if the last prediction frame is a *motion frame* which contains motion regions; or do nothing (*meaning that this frame is a still frame without motion objects*), otherwise.
2. For each motion region R_i within R , perform the following steps.
 - 2.1 For each macroblock MB in R_i , perform the following steps.
 - 2.1.1 If the prediction mode of MB is B_PRED, then set the *authentication score* to 1.
 - 2.1.2 If the prediction mode of MB is H_PRED, V_PRED, DC_PRED, or TM_PRED, then set the authentication score to 5.
 - 2.2 Compute a total value T_s of the authentication scores of R_i .

- 2.3 Record the index of F , the index of R_i , and T_s into S_a .
3. Repeat Step 2 until all motion regions in R are processed.

Most contents of the frames of surveillance videos are backgrounds and contain no motion objects. When finding motion objects in a key frame, the proposed method uses Algorithm 4.2 to generate authentication signals. In Step 1 of Algorithm 4.2, if the last prediction frame is a motion frame, then we analyze further the motion regions and generate authentication signals for two reasons. First, if motion objects exist in the last prediction frame, motion objects will not have huge move in the next frame in general. Second, we can not detect motion objects by prediction modes or motion vectors in key frames due to the prediction process. So, by using the motion regions in the last prediction frame, in Step 2 we analyze the prediction modes in each motion region to generate authentication scores as part of the authentication signals.

4.3 Embedding and Extracting of Authentication Signals in Surveillance Videos

In this section, the proposed methods of embedding and extracting authentication signals are introduced. In Section 4.3.1, the proposed technique of embedding authentication signals is described, and in Section 4.3.2 the proposed technique of extracting authentication signals is presented.

4.3.1 Embedding of Authentication Signals

In this section, the proposed technique of embedding authentication signals is described. In Section 4.3.1.1, the proposed idea is presented. In Section 4.3.1.2, the

detailed steps of the embedding process are described.

4.3.1.1 Proposed Idea

Each authentication signal generated by the way described in Section 4.2 has a uniform structure to be embedded into videos, which can be divided to four parts by the proposed data hiding method. Each part has a unique mark to know which part it is. Also, the authentication signals are duplicated into several copies in order to reduce the probability of misrepresentation when extracting authentication signals in the extraction process.

Furthermore, we modify the frequency coefficients of the chroma color space mentioned in Chapter 3 to embed authentication signals. A flowchart of the process for embedding authentication signals is illustrated in Figure 4.5 and a detailed algorithm is described in Section 4.3.1.2.

4.3.1.2 Process of embedding authentication signals

After obtaining the authentication signals from the authentication signal generation process mentioned above, we duplicate each authentication signal into several copies. The total number of these copies is smaller than the data hiding capacity of the surveillance video. The main purpose of this duplication process is to facilitate extracting authentication signals more precisely using a voting process in the later extraction process to reduce the probability of misrepresentation. Then, the authentication signals and the duplication time are embedded into the prediction frame using the data hiding method which is similar to the one mentioned previously in Chapter 3. The details are described as an algorithm as follows.

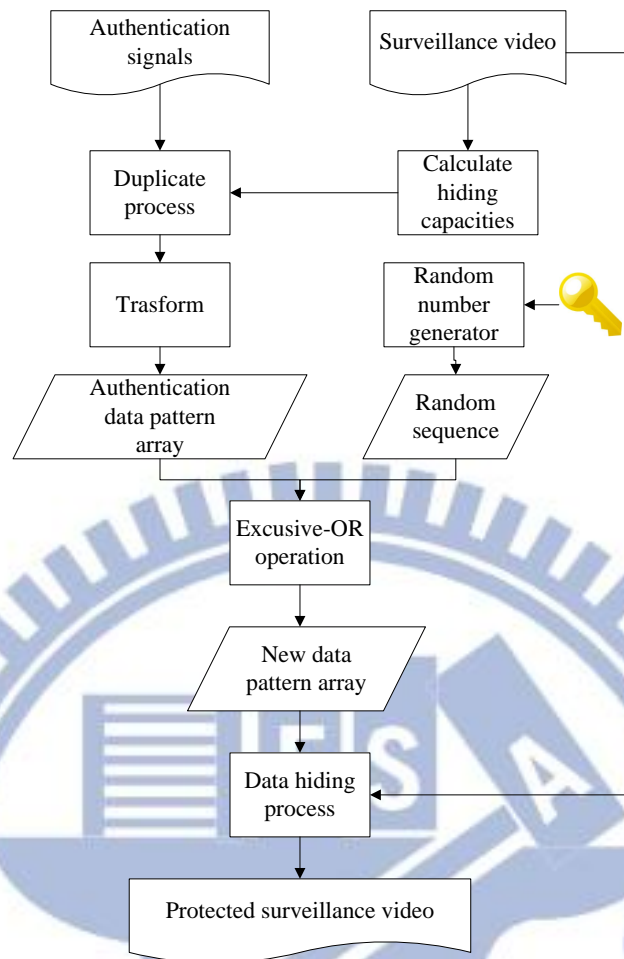


Figure 4.5 A flowchart of the process for embedding authentication signals.

Algorithm 4.3 Process for embedding authentication signals.

Input: a surveillance video V , a secret key K_s , a report E of authentication signals, and a random number generator f .

Output: a protected WebM video.

Steps.

Stage 1 --- initialization.

1. Detect V to get a number C of the hiding capacity in this video by perform Algorithm 3.1.
2. Calculate the total amount A_u of authentication signals in E .
3. Duplicate each authentication signal in E $K - 1$ times, concatenate them and transform the result into a binary string B , where K is calculated by:

$$K = \frac{N}{4 \times A_u}. \quad (4.4)$$

4. Combine the duplication time $(K - 1)$ with an *extraction mark* and transform the result into a binary string S_k .
5. Duplicate S_k $K - 1$ times and append the result to B .
6. Divide B into a linear array A of 4-bit segments.
7. Use the secret key K_s as a seed to generate a sequence Q of random numbers using the random number generator f ; and take the first eight random numbers from Q , denoted as N_1 through N_8 .

Stage 2 --- embedding message data into the video.

8. Perform the following steps to embed the authentication signals S_b into each *unprocessed* macroblock MB in every prediction frame of V , assuming V is large enough to embed the entire message.

- 8.1 Take eight unembedded 4-bit elements from A , denoted as A_1 through A_8 , with each element A_i corresponding to a data pattern denoted as P_i .
- 8.2 Combine P_i with N_i by exclusive-OR operations to form a new data pattern DP_i in the following way:

$$DP_i = P_i \oplus N_i, \quad (4.5)$$

where $i = 1, 2, \dots, 8$.

- 8.3 Save the original quantized coefficients of the chroma color channels in MB .
- 8.4 (*Embedding the secret data patterns*) Check the coefficients of each of the eight corresponding subblocks of the chroma color channels (four in the U channel and the other four in the V channel), denoted as SB_i , $i = 1, 2, \dots, 8$: if the original coefficients of SB_i do not match those of data pattern DP_i , then modify them to be so by changing those mismatching ones in SB_i to be the corresponding ones of data pattern DP_i ; else, do nothing.

8.5 (*Computing the resulting distortion*) Calculate the mean square quantization error (MSQE) between the content MB_o of the original macroblock MB and the content MB_o' of the modified macroblock MB' of the chroma color channels ($i = 1, 2$ for the U channel and the V channel, respectively) as follows:

$$MSQE_i = \sqrt{(MB_o' - MB_o)^2} \quad (4.6)$$

where each of MB_o and MB_o' means an 8×8 vector of coefficients.

8.6 Calculate the following value of the peak signal-to-noise ratio (PSNR) $PSNR_i$ of the chroma color channels ($i = 1, 2$):

$$PSNR_i = 10 \times \log \left(\frac{S_{peak}^2}{MSQE_i} \right) \quad (4.7)$$

where S_{peak} means the maximum possible pixel value of the image.

8.7 Calculate the average PSNR value $PSNR_{avg}$ of the chroma color channels ($i = 1, 2$):

$$PSNR_{avg} = \frac{\sum_i PSNR_i}{2} \quad (4.8)$$

8.8 (*Checking the data embeddability of the macroblock*) Use the ROI map index to label the modified macroblock MB' by setting its value to be 1 (meaning that macroblock MB is *data-embeddable*) if $PSNR_{avg}$ is smaller than the pre-selected threshold T ; else, use the default value of the ROI map index which is 0 and recover the modified coefficients of the chroma color channels in MB to get their original values (meaning that macroblock MB is *non-data-embeddable* and the eight 4-bit elements from A , i.e. A_1 through A_8 have not been embedded).

9. Repeat Step 8 until the entire secret message in array A has been embedded (i.e., until all elements in A have been processed).

In Step 8.9, if the ROI map index is set as 0, it means that modifications of the coefficients of the chroma color channels will cause too much distortion in the resulting image, allowing one to perceive any modification that has been done, so that imperceptibility would not be achieved. Therefore, we need to recover the modified coefficients to their original values.

4.3.2 Extraction of Authentication Signals

In this section, the proposed technique of extracting authentication signals is described. In Section 4.3.2.1, the proposed idea is presented, and in Section 4.3.2.2, the detail steps of the extraction process are described.

4.3.2.1 Proposed Idea

We extract authentication signals hidden in a video by a data extraction method which is similar to the one mentioned previously in Chapter 3, and get authentication signals based on the use of a *voting process*. As mentioned in Section 4.3.1, we have duplicated authentication signals into several copies, so we need to get the duplication time before conducting the voting process using the authentication signals. A flowchart of the process for authentication signal extraction is shown in Figure 4.6. The corresponding detailed algorithm is described in Section 4.3.2.2.

4.3.2.2 Process of extracting authentication signals

In the previous embedding process of authentication signals, we duplicated the authentication signals several times. Because sometimes the recompression will cause slight changes of the frequency coefficients, some of the embedded authentication signals may be destroyed. Therefore, we can extract the authentication signals more

precisely by the voting process if the tampered area is not too large and the duplication time is large enough.

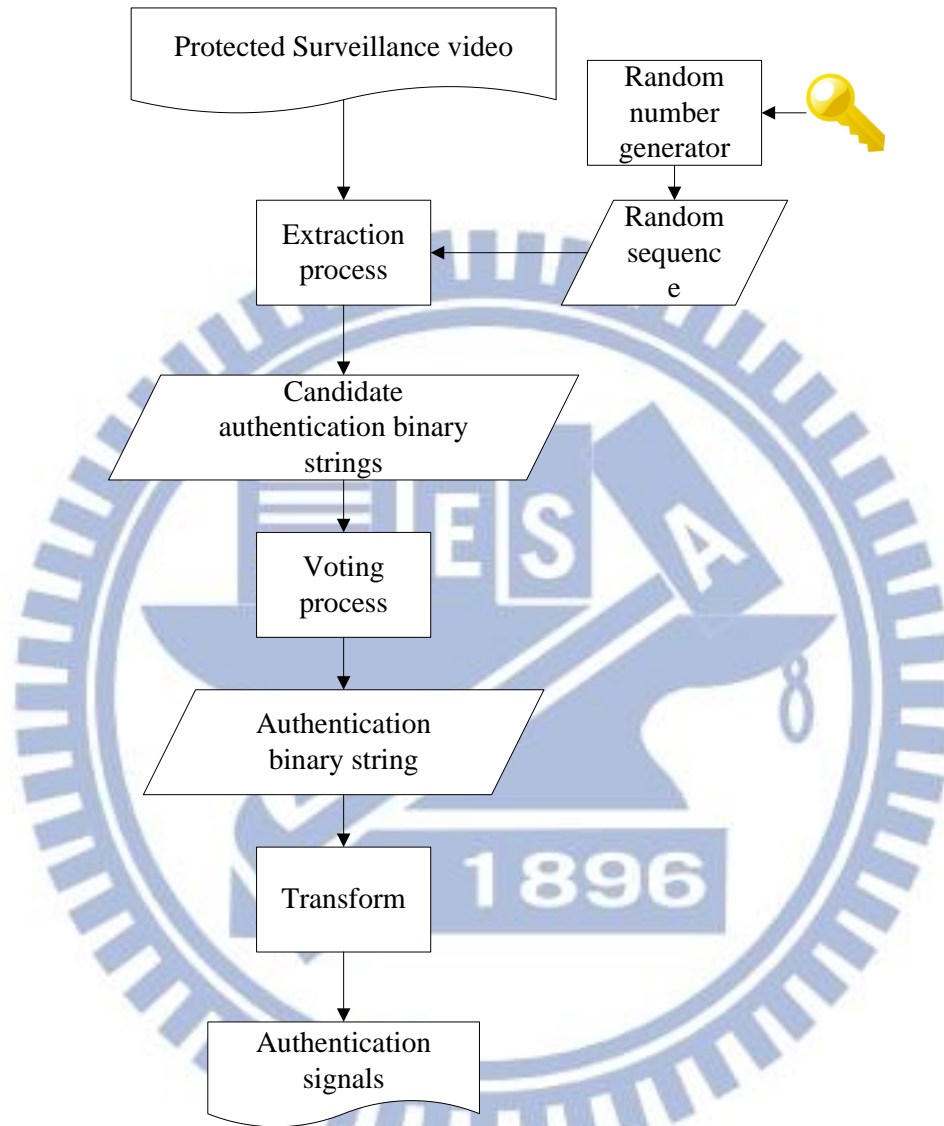


Figure 4.6 A flowchart of the process for authentication signal extraction.

Each authentication signal is embedded in exactly four macroblocks in the proposed hiding method. The extraction method extracts data from one macroblock at a time, so an authentication signal needs four times to extract. As a result, an authentication signal is divided into four parts, each part having a unique mark for the extraction process to recognize which part the extracted result is. Therefore, we need

four candidate authentication signal buffers to store the four parts of the candidate authentication signals for use in the voting process. In the proposed voting process, each bit of the extracted data may be either of the two possible values, 0 and 1, so every bit b_i of the extracted candidate authentication signal is associated with two scores: Score-0 and Score-1. If the value of b_i is 0, one vote is added to Score-0; otherwise, to Score-1. Then, the value with the higher vote score will be regarded as the correct value of b_i . As a result, after all candidate authentication signals are processed, we can get the correct authentication signals according to the voting result.

Algorithm 4.4 Process for extracting authentication signals.

Input: a protected surveillance video V , a secret key K_s , and a random number generator f .

Output: an authentication report E of V .

Steps.

Stage 1 --- extracting message data from the input video.

1. Use the secret key K_s as a seed for f , use f to generate a sequence of random numbers, Q , and take the first eight random numbers from Q , denoted as N_1 through N_8 .
2. Perform the following steps to extract the authentication signals from each unprocessed macroblock MB in every prediction frame of V .
 - 2.1 Check the coefficients of each of the eight corresponding subblocks of the chroma color channels in MB (four in the U channel and the other four in the V channel), denoted as SB_j , $j = 1, 2, \dots, 8$: if the coefficients of SB_j match those of data pattern i , then record the code of *data pattern* i as an *extracted result* DP_i .
 - 2.2 Combine DP_i with N_i by exclusive-OR operations to become a new data

pattern P_i :

$$P_i = DP_i \oplus N_i \quad (4.9)$$

where $i = 1, 2, \dots, 8$.

- 2.3 Transform P_i ($i = 1, 2, \dots, 8$) into a 4-bit binary string S_i and concatenate them to form a binary string $S = S_1S_2S_3S_4S_5S_6S_7S_8$.

Stage 2 --- analyzing extracted message data.

- 2.4 Analyze S according to the following rules:

- (1) if the first 8-bits of S are 00010001, put S into the candidate authentication signal buffer of part 1 (meaning that S is a candidate authentication signal of part 1);
- (2) if the last 8-bits of S are 00010010, put S into the candidate authentication signal buffer of part 2;
- (3) if the last 8-bits of S are 00010011, put S into the candidate authentication signals buffer of part 3;
- (4) if the last 8-bits of S are 00010100, put S into the candidate authentication signals buffer of part 4;
- (5) if the last 16-bits of S are 0001001000110100, put S into the *duplication time buffer* (meaning that S is a candidate duplication time);
- (6) otherwise, drop S .

3. Repeat Step 2 until all macroblocks in V are processed.

Stage 3 ---process for getting the duplication time.

4. Perform the following steps to get the duplication time K .

- 4.1 Denote each binary string in the duplication time buffer as $S_i = b_1b_2b_3\dots b_l$, where $1 \leq i \leq n$ with n being the total number of elements in the duplication time buffer and l being the length of S_i .

4.2 Associate each bit of S_i with two vote scores $V_0[m]$ and $V_1[m]$, where $1 \leq m \leq l$.

4.3 Calculate the score of each bit of S_i according to the following rule, where $1 \leq i \leq n$:

$$\begin{aligned} & \text{if } b_m = 0, \text{ then set } V_0[m] = V_0[m] + 1; \\ & \text{if } b_m = 1, \text{ then set } V_1[m] = V_1[m] + 1, \end{aligned} \quad (4.10)$$

where $1 \leq m \leq l$.

4.4 Denote $S_v = s_1 s_2 s_3 \dots s_l$, and compute s_i in S_v by the following rule:

$$\begin{aligned} & \text{if } V_0[m] > V_1[m], \text{ then set } s_m = 0; \\ & \text{if } V_1[m] > V_0[m], \text{ then set } s_m = 1, \end{aligned} \quad (4.11)$$

where $1 \leq m \leq l$.

4.5 Transform the first 16-bits of S_v to be K (meaning K is the duplication time).

Stage 4 ---process for voting by the candidate authentication signals to get the authentication signal.

5. Take K unprocessed elements from the candidate authentication signal buffer to perform the following steps to get each part of authentication signals, RS_i ($i = 1, 2, 3, 4$).

5.1 Divide K elements into halves, the former one being a set A_1 , and the latter another set A_2 , each with $K/2$ elements, to perform the following steps.

5.1.1 Denote each element in A_j as $S_i = b_1 b_2 b_3 \dots b_l$, where $1 \leq i \leq n$ with n being the total elements in A_j and l being the length of S_i .

5.1.2 Associate each bit of S_i with two vote scores $V_0[m]$ and $V_1[m]$, where $1 \leq m \leq l$ and the score of each bit of S_i is computed according to Eq. 4.10.

5.1.3 Denote $SA_j = s_1 s_2 s_3 \dots s_l$, ($j = 1, 2$) according to Eq. 4.11, where $1 \leq m \leq l$.

6. Save SA_1 as RS_i and mark the K elements as processed, if SA_1 is equal to SA_2 ; or take elements in A_1 to perform Step 5.1 until the number of elements in A_1 is less than a pre-defined value T and mark those elements as processed, otherwise.
7. Combine RS_i ($i=1, 2, 3, 4$) to form an authentication signal and record it into E .
8. Repeat Steps 5 and 6 until all elements in candidate binary string buffers are processed.

4.4 Authentication of Surveillance Videos

In this section, the proposed techniques for detection and verification of video tampering, which use the authentication signals extracted in a way as described in Section 4.3.2, are introduced. Each frame in a protected video is treated as a unit of authentication if there are motion objects detected in the frame by the proposed motion detection method. For these suspected frames, we perform authentication on them to get more information about the tampering, which will be described later. In Section 4.4.1, the process for detection and verification of tampering in key frames is described. And the process for detection and verification of tampering in prediction frames is presented in Section 4.4.2.

4.4.1 Detection and Verification of Tampering in Key Frames

During the process for generation of authentication signals for a key frame K , if the last prediction frame is a motion frame which has motion regions, then we analyze the prediction modes of macroblocks in each motion region in order to generate a

score and denote the score as an authentication signal in this key frame. When we have to authenticate a key frame in a protected surveillance video, we can analyze the prediction mode information to get scores about authentication regions to know whether there exist some motion objects missing in K . The details are as follows.

Algorithm 4.5 Process for detection of tampering in key frames.

Input: a suspicious surveillance video V and an authentication report E of V .

Output: authenticated key frames with information of missing motion objects in V .

Steps.

1. For each key frame F in V , perform the following steps.
 - 1.1 Extract each authentication signal S_i in E , if the index of F in S_i is equal to the index of F , and denote the extracted authentication signals as S .
 - 1.2 For each authentication signal S_i in S , perform the following steps.
 - 1.2.1 Extract the motion region information from the last prediction frame according to the index of R_i in S_i as the *authentication region* RA_i in F .
 - 1.2.2 Check prediction modes of each macroblock MB within RA_i by the following steps.
 - (1) If the prediction mode of MB is B_PRED, then set the *score* to 1.
 - (2) If the prediction mode of MB is H_PRED, V_PRED, DC_PRED, or TM_PRED, then set the score to 5.
 - (3) Compute a total scores S_i of the scores of RA_i .
 - 1.2.3 Compare the score between the score of authentication signal in S_i and S_i ; if the score is different, then mark the authentication region RA_i as information of missing a motion object in F .
 - 1.3 Repeat Step 1.2 until all authentication signals in S are verified for F .
2. Repeat Step 1 until reaching the end of V .

4.4.2 Detection and Verification of Tampering in Prediction Frames

During the process for generation of authentication signals of a prediction frame F , the authentication signals are composed of motion regions in F . These signals are formed based on the prediction mode information and the motion vectors of the corresponding motion regions in F ; therefore, the authentication signals contain some information about the motion objects. When we need to authenticate a prediction frame in a protected surveillance video, we can analyze the prediction mode information of the motion regions to get more information about this region to know whether there exist some motion objects missing in F .

Algorithm 4.6 Process for detection of tampering in prediction frames.

Input: a suspicious surveillance video V and an authentication report E of V .

Output: authenticated prediction frames with information of missing motion objects in V .

Steps.

1. For each prediction frame P in V , perform the following steps.
 - 1.1 Extract each authentication signal S_i in E , if the index of S_i is equal to the index of P ; and denote the extracted authentication signals as S .
 - 1.2 For each authentication signal S_i in S , perform the following steps.
 - 1.2.1 Extract motion region information from S_i , and denote it as R_m .
 - 1.2.2 For each macroblock MB in R_m , perform the following steps.
 - (1) If the prediction mode of MB is intra coded such as DC_PRED, H_PRED, V_PRED, TM_PRED, and B_PRED, then set the *motion-region score* to be 5.

- (2) If the prediction mode of MB is inter coded such as NEWMV, NEARMV, NEARESTMV, and SPLITMV, then set the motion-region score to be 10; otherwise, set the motion-region score to be -10 .
- (3) Compute the total motion-region score S_t of the motion-region scores in R_m ; and if S_t is negative, then mark the R_m as information of missing a motion object in P .

1.3 Repeat Step 1.2 until all authentication signal S_i in S have been verified for P .

2. Repeat Step 1 until reaching the end of V .

In Step 3, based on the motion detection technique mentioned in Section 4.2, a motion object grouped into a motion region will cover almost the entire motion region. Sometimes a macroblock whose prediction mode is the ZEROMV mode may be contained in a motion region, but would not be the majority. So, if the total motion-region score S_t of a motion region is negative, it means that a motion object is already missed in the motion region. Therefore, we mark this motion region as the authentication result.

4.5 Experimental Results

In our experiments, an WebM surveillance video whose format of each video frame is CIF(352×288) was used as the input. Four results of the test videos are shown below. Let them be denoted as test1, test2, test3, and test4. Six frames of the original video of test1 are shown in Figure 4.7. Six corresponding frames of the protected video of test1 after performing the proposed authentication signal embedding process

are shown in Figure 4.8. Figure 4.9 shows an imperceptible result of tampering on the six frames by taking part of the background image to cover the walking person in the video of test1 to remove him. Figure 4.10 is the authentication result of these tampered frames, in which the green areas represent motion objects missing in these frames. The corresponding results of other three tested videos are shown in Figures 4.11 through Figures 4.16.

During the process of authentication signals extraction, some authentication signals may get lost. In Figure 4.16, although the motion objects in the 141th and 146th frame are missing, we can not verify them by the corresponding authentication signals because the authentication signals were lost. Furthermore, we took the video of test1 to test how the tampered region size in a frame and the tampered number of frames influence the missing rate of the authentication signals. Table 4.1 shows the missing rate of the authentication signals of the test1 video. As shown in Table 4.1, if the tampered percent of a region in a frame is too large (in test1 video the percent is larger than 0.67), the number of lost authentication signals will be affected by the tampered frames more obviously.

4.6 Discussions and Summary

In this chapter, we have proposed an authentication method which can detect motion objects in surveillance videos in order to generate authentication signals and verify tampering in a protected surveillance video. The proposed method uses the prediction mode information and motion vectors in WebM videos to generate authentication signals and embeds authentication signals into prediction frames of the input video. To extract the authentication signals more precisely, we use the voting technique to make sure we can still extract the correct authentication signals while

regions of a suspicious frame are not tampered with largely. The correct authentication signals are used to detect tampering and verify the region contents in a protected surveillance video. The proposed authentication system not only checks if a protected video has been tampered with or not, but also further shows where and how the tampering occurs.

Table 4.1 Experimental results of loss rate of authentication signals of test1 video (total number of frames in test1 video is 270, and total number of authentication signals in test1 video is 87).

Tampered number of frames	Percent of tampered region in a frame (%)	Number of lost authentication signals.	Percent of total tampered area in video (%)	Loss rate of authentication signals (%)
90	0.67	7	0.215	0.08
90	0.75	24	0.25	0.275
90	1	25	0.33334	0.287
135	0.67	12	0.335	0.137
135	0.75	42	0.375	0.482
135	1	46	0.5	0.582
180	0.67	19	0.44	0.218
180	0.75	54	0.5	0.62
180	1	81	0.66667	0.931



Figure 4.7 Six frames of the original video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame.



(c)



(d)



(e)



(f)

Figure 4.7 Six frames of the original video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame (cont'd).



(a)



(b)

Figure 4.8 Six frames of the protected video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame.



(c)



(d)



(e)



(f)

Figure 4.8 Six frames of the protected video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame (cont'd).



(a)



(b)

Figure 4.9 Six frames of the tampered video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame.



(c)

(d)



(e)

(f)

Figure 4.9 Six frames of the tampered video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame (cont'd).



(a)

(b)

Figure 4.10 Six frames of the authenticated video of test1. (a) The 152th frame (b) The 153th frame (c)

The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame.

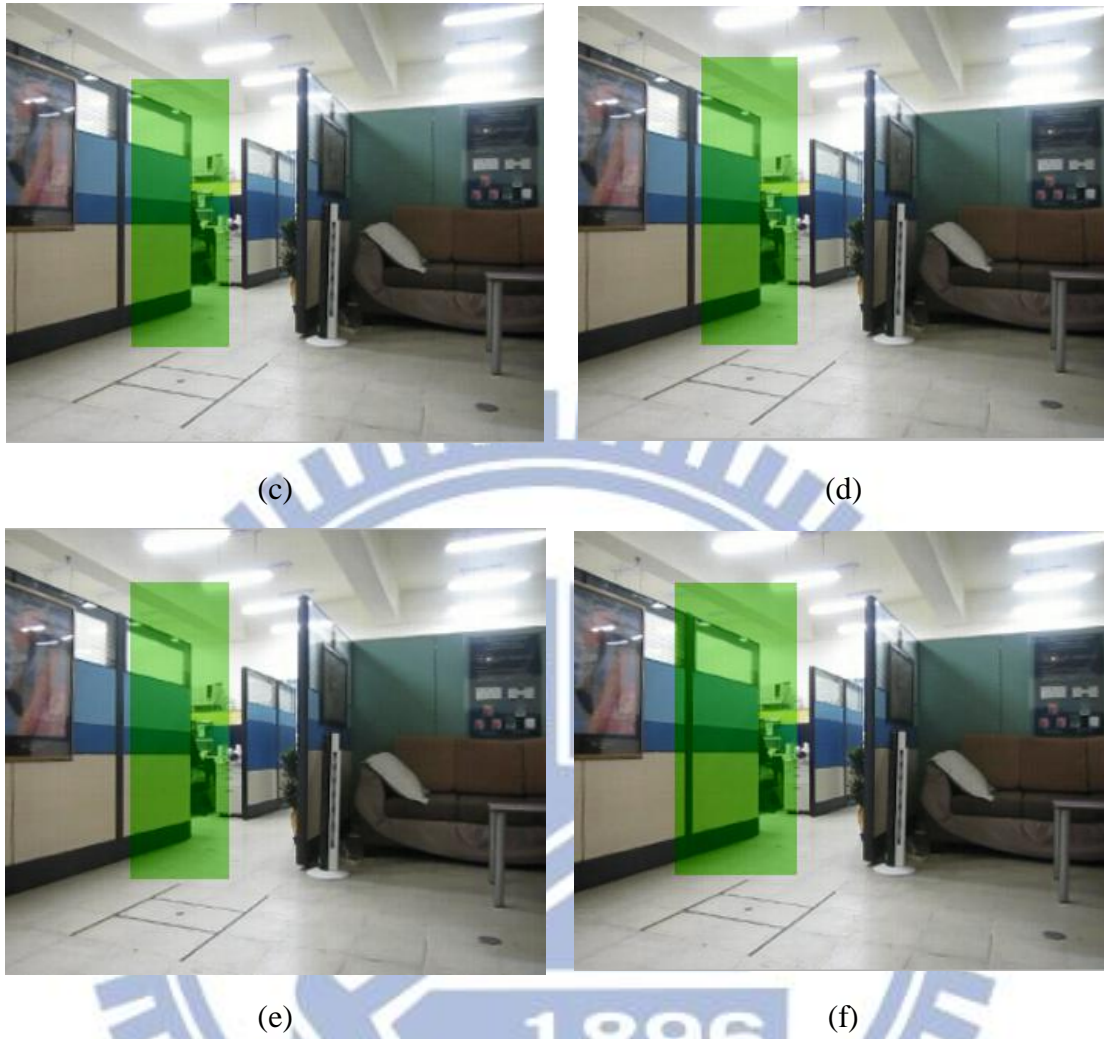


Figure 4.10 Six frames of the authenticated video of test1. (a) The 152th frame (b) The 153th frame (c) The 154th frame (d) The 155th frame (e) The 156th frame (f) The 157th frame (cont'd).



Figure 4.11 Six frames of the protected video of test2. (a) The 48th frame (b) The 49th frame (c) The 50th frame (d) The 51th frame (e) The 52th frame (f) The 53th frame.



(c)



(d)



(e)

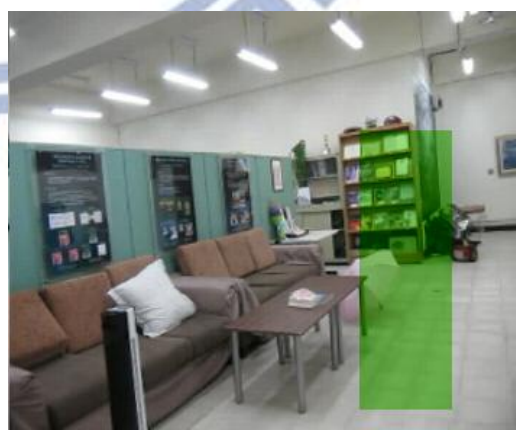


(f)

Figure 4.11 Six frames of the protected video of test2. (a) The 48th frame (b) The 49th frame (c) The 50th frame (d) The 51th frame (e) The 52th frame (f) The 53th frame (cont'd).



(a)



(b)

Figure 4.12 Six frames of the authenticated video of test2. (a) The 48th frame (b) The 49th frame (c) The 50th frame (d) The 51th frame (e) The 52th frame (f) The 53th frame.



(c)

(d)



(e)

(f)

Figure 4.12 Six frames of the authenticated video of test2. (a) The 48th frame (b) The 49th frame (c) The 50th frame (d) The 51th frame (e) The 52th frame (f) The 53th frame (cont'd).



(a)

(b)

Figure 4.13 Six frames of the protected video of test3. (a) The 68th frame (b) The 69th frame (c) The 70th frame (d) The 71th frame (e) The 72th frame (f) The 73th frame.



(c)

(d)



(e)

(f)

Figure 4.13 Six frames of the protected video of test3. (a) The 68th frame (b) The 69th frame (c) The 70th frame (d) The 71th frame (e) The 72th frame (f) The 73th frame (cont'd).



(a)

(b)

Figure 4.14 Six frames of the authenticated video of test3. (a) The 68th frame (b) The 69th frame (c) The 70th frame (d) The 71th frame (e) The 72th frame (f) The 73th frame.



(c)



(d)



(e)



(f)

Figure 4.14 Six frames of the authenticated video of test3. (a) The 68th frame (b) The 69th frame (c) The 70th frame (d) The 71th frame (e) The 72th frame (f) The 73th frame (cont'd).



(a)



(b)

Figure 4.15 Six frames of the protected video of test4. (a) The 141th frame (b) The 142th frame (c) The 143th frame (d) The 144th frame (e) The 145th frame (f) The 146th frame.



(c)

(d)



(e)

(f)

Figure 4.15 Six frames of the protected video of test4. (a) The 141th frame (b) The 142th frame (c) The 143th frame (d) The 144th frame (e) The 145th frame (f) The 146th frame (cont'd).



(a)

(b)

Figure 4.16 Six frames of the authenticated video of test4. (a) The 141th frame (b) The 142th frame (c) The 143th frame (d) The 144th frame (e) The 145th frame (f) The 146th frame.



(c)



(d)



(e)



(f)

Figure 4.16 Six frames of the authenticated video of test4. (a) The 141th frame (b) The 142th frame (c) The 143th frame (d) The 144th frame (e) The 145th frame (f) The 146th frame (cont'd).

Chapter 5

Protection of Privacy-sensitive Contents in Surveillance Videos Using WebM Video Features

5.1 Introduction

Privacy protection is a very important issue in video surveillance. Since video surveillance systems exist everywhere in our environment and usually conduct space monitoring for long time periods, it may record information of individuals and so violate protection of personal privacy. Hence, it is necessary in some cases to hide the privacy-violating parts of surveillance video contents to avoid legal disputes and to protect personal privacy from being misused by suspicious people. Therefore, we propose a method for privacy protection to solve this issue and the method is described in this chapter.

In Section 5.1.1, the related problem definitions are given. In Section 5.1.2, the idea of the proposed method is described. In Section 5.2, the proposed process for hiding the privacy-sensitive contents of a privacy region in videos is presented. In Section 5.3, the proposed process for recovering the privacy-sensitive contents of a privacy region from videos is presented. Some experimental results are shown in the Section 5.4. Finally, some discussions and a summary will be given in the last section of this chapter.

5.1.1 Problem Definition

In the privacy protection problem dealt with in this study, an authorized user can specify a region R to be protected in an input video. If there exist privacy-sensitive contents in R , which need to be protected, then a process of removing and replacing them with the background image in order not to reveal the privacy-sensitive contents of R will be conducted automatically. Also, the privacy-sensitive contents of R are hidden into the video to produce a *privacy video*. Thereafter, once the privacy-sensitive contents need to be recovered, the data hidden in the privacy video is extracted and used to recover the privacy-sensitive contents of R .

Two main issues are involved in this problem. The first is how to replace the video contents of R with the background image and to embed enough information about the contents of R into the video for use in the recovery stage. The second is how to extract the data from the privacy-protected video and to recover the original contents of the protected region.

5.1.2 Proposed Ideas

Like other video codecs such as the H.264, the VP8 video codec has a process to find the best prediction block in blocks. A *motion vector* is used to indicate the location of the best prediction block. The difference between the best prediction block and the currently-processed block is DCT-based transformed into a set of *frequency coefficients*. Motion vectors and frequency coefficients are used in the decoding process to decode the corresponding block, called the *decoding information* here.

A video can be decoded correctly based on the decoding information generated during the encoding process. The idea we propose to protect a privacy-sensitive region R which is specified by an authorized user is to set the decoding information of

R to be some pre-defined values. In this way, the video contents can be removed and replaced with the background image. The decoding information of R is then hidden into the input video. If the privacy-sensitive contents of R need be recovered, the decoding information of R hidden in the video is extracted and used to conduct the recovery work.

5.2 Hiding of Privacy-sensitive Contents

In this section, the proposed process for removing and hiding privacy-sensitive contents is introduced. In Section 5.2.1, the principle of the proposed method is given. In Section 5.2.2, the proposed process for hiding privacy-sensitive contents is described.

5.2.1 Principle of Proposed Method

The most important problem we need to overcome is how to replace the privacy-sensitive contents in an area with the background image without any negative effects. In order to remove privacy-sensitive contents in the user-specified region R , we need to replace an *encoded macroblock* in the encoding process, but this may cause a problem; called the *reference problem* here. This problem occurs when an encoded macroblock is used as a reference to encode other macroblocks which have not been encoded yet during the encoding process, so that when we modify the motion vectors and frequency coefficients in an encoded macroblock MB in order to remove privacy-sensitive contents, the macroblocks which refer to the encoded macroblock MB will cause errors in the decoding result. Figure 5.1 shows an example of the reference problem.



Figure 5.1 An example of reference error.

Other video codecs such as H.264 has a feature called *multiple slice groups* which can control the encoding order of each slice groups by setting the slice group identifier to avoid the reference problem. However, the VP8 encoder encodes macroblocks only in a raster-scan order, so it can not change encoding order to solve the reference problem. As a result, we propose using the *golden reference frame* which is a feature of the WebM video already mentioned in Section 2.5.5 to solve the reference problem. Because a surveillance often comes from monitor a fixed area for a long time and the background image is usually still with no moving object included, using the golden reference frame not only can solve the reference problem but also can take the efficiency of the compression rate into consideration.

In the VP8 video codec, a type of encoded macroblock is called the intra-coded macroblock whose prediction mode is the intra prediction mode. A feature of the intra-coded macroblock is that it does not use any reference frames, such as the last frame, the golden reference frame, and the alternative reference frame. Therefore, if we modify the frequency coefficients in an intra-coded macroblock in order to remove the privacy-sensitive contents for protecting the personal privacy within R , it will result in a grey color macroblock rather than the background image. Figure 5.2 shows

an example of the above problem. In Section 5.2.2 we will have more illustrations about how to solve this problem.



Figure 5.2 An example of modify the frequency coefficients in an intra-coded macroblock to yield intra-coded grey macroblocks.

5.2.2 Process for Hiding Privacy-sensitive Contents

The proposed process is applied to the prediction frames of an input WebM video. If a region R specified by an authorized user has privacy-sensitive contents which need to be protected, the motion vectors and frequency coefficients of the currently-processed macroblock within R is all set to zero. Therefore, the video contents of R become the corresponding part of the background image which has appeared in the golden reference frame of the input video. It also places a restriction on this proposed process that the first frame of the input video must be a background frame. The values of the original motion vectors and frequency coefficients of the macroblocks of R are grouped into a report and then hidden into the prediction frames in the input video. Furthermore, we use a secret key to randomize the report for the security protection issue.

As mentioned before, macroblocks are encoded in a raster-scan order in the VP8 video codec. So, we have to perform an encoding process to remove the privacy-sensitive contents and get the decoding information which may be used to recover the privacy-sensitive contents if necessary, and then perform another encoding process to hide the decoding information into the video. The following algorithm describes the details of the above-mentioned process.

Algorithm 5.1 Process for removing the privacy-sensitive contents in a user-specified region.

Input: a WebM video V and a region R specified by an authorized user

Output: a WebM video V' with privacy-sensitive contents in R removed and a report E of privacy-sensitive contents

Steps.

Stage 1 --- initialization.

1. Set the *golden reference frame* as the reference frame for each prediction frame F in V .
2. Restrict the encoder to use inter prediction modes during the prediction step.

Stage 2 --- removing privacy-sensitive contents in the region R .

3. For each prediction frame F in V , perform the following steps.
 - 3.1 Detect motions in the region R of F by checking the prediction mode of each macroblock within R ; set the value of the motion flag to be 1, if there exist prediction modes other than ZEROMV within R ; otherwise, set the value of the motion flag to be 0.
 - 3.2 If the value of motion flag is set as 1, then for each macroblock MB within R of F , perform the following steps.
 - 3.2.1 For each subblock in MB , perform the following steps.

- (1) Record all the sixteen coefficients of the subblock of Y2 into E and set all coefficients of the Y2 block to be zero.
- (2) Record the DC coefficient of each subblock of the chroma color channels (including the U channel and the V channel) into E and set all coefficients of these subblocks to be zero.
- (3) Record seven coefficients of each subblock of the luma color channel (the Y channel) into E and set all coefficients of these subblocks to zero.
- (4) Record the index of MB and the index of F into E .

3.2.2 Record the motion vector of MB into E and set the motion vector to be zero.

4. Repeat Step 3 until all frames of V have been processed.

Considering the capacities of the proposed hiding data method, we cannot record all the coefficients in a macroblock which will be used to recover privacy-sensitive contents. Therefore, we conduct some tests in order to decide which coefficients of color channels need to be recorded. In Section 2.5.1, we mentioned that there exists a 4×4 subblock called Y2 which records the DC coefficients of a Y block. If we lose Y2 coefficients, we cannot recover the contents in this macroblock. Figure 5.3 shows an example if we lose the Y2 coefficients information. Therefore, we record all Y2 coefficients information in order to make sure we can recover the privacy-sensitive contents. In Step 3.2.1, because the VP8 video codec uses the zig-zag scan order, which is shown in Figure 5.4, to encode subblocks, after our experimental tests we decide to record seven coefficients of each Y subblock according to the zig-zag scan order.



Figure 5.3 Comparison between the original image and the image whose Y2 coefficients have been lost. (Left) the original image. (Right) the image whose Y2 coefficients have been lost.

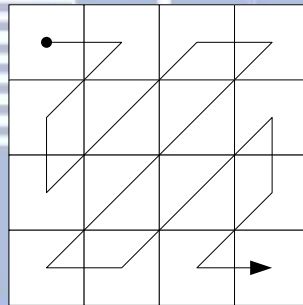


Figure 5.4 The zig-zag scan order.

Algorithm 5.2 Process for embedding the privacy-sensitive contents into a WebM video.

Input: a WebM video V with privacy-sensitive contents in R removed, a secret key K , a random number generator f , and a report E of the privacy-sensitive contents.

Output: a WebM video V' with privacy-sensitive contents in R removed and embedded.

Steps.

Stage 1 --- initialization.

1. (Randomizing the message) Transform the report E into a binary string B , use the secret key K as a seed to generate a sequence Q_r of random numbers using the

random number generator f , and randomize B with Q_r to get a randomized binary string B' .

2. Divide B' into a linear array A of 4-bit segments.
3. Use the secret key K as a seed to generate a sequence Q of random numbers using the random number generator f ; and take the first eight random numbers from Q , denoted as N_1 through N_8 .

Stage 2 --- embedding message data into the video.

4. Perform the following steps to embed the input report E into each *unprocessed* macroblock MB in every prediction frame of V , assuming V is large enough to embed the entire message.

4.1 Take eight unprocessed 4-bit elements from A , denoted as A_1 through A_8 , with each element A_i corresponding to a data pattern denoted as P_i (e.g., if $A_i = 1001_2 = 9_{10}$, then $P_i = \text{data pattern } 9$).

4.2 Combine P_i with N_i by exclusive-OR operations to become a new data pattern DP_i :

$$DP_i = P_i \oplus N_i \quad (5.1)$$

where $i = 1, 2, \dots, 8$.

4.3 Save the original quantized coefficients of the chroma color channels in MB .

4.4 (*Embedding the secret data patterns*) Check the coefficients of each of the eight corresponding subblocks of the chroma color channels (four in the U channel and the other four in the V channel), denoted as SB_i , $i = 1, 2, \dots, 8$; if the original coefficients of SB_i do not match those of data pattern DP_i , then modify them to be so by changing those mismatching ones in SB_i to be the corresponding ones of data pattern DP_i ; else, do nothing.

4.5 (*Computing the resulting distortion*) Calculate the mean square quantization

error (MSQE) between the content MB_o of the original macroblock MB and the content MB_o' of the modified macroblock MB' of the chroma color channels ($i=1, 2$ for the U channel and the V channel, respectively):

$$MSQE_i = \sqrt{(MB_o' - MB_o)^2} \quad (5.2)$$

where each of MB_o and MB_o' means an 8×8 vector of coefficients.

- 4.6 Calculate the following value of the peak signal-to-noise ratio (PSNR) $PSNR_i$ of the chroma color channels ($i = 1, 2$):

$$PSNR_i = 10 \times \log \left(\frac{S_{peak}^2}{MSQE_i} \right) \quad (5.3)$$

where S_{peak} means the maximum possible pixel value of the image.

- 4.7 Calculate the average PSNR value $PSNR_{avg}$ of the chroma color channels ($i = 1, 2$):

$$PSNR_{avg} = \frac{\sum_i PSNR_i}{2}. \quad (5.4)$$

- 4.8 (*Checking the data embeddability of the macroblock*) Use the ROI map index to label the modified macroblock MB' by setting its value to be 1 (meaning that macroblock MB is *data-embeddable*) if $PSNR_{avg}$ is smaller than the pre-selected threshold T ; else, use the default value of the ROI map index which is 0 and recover the modified coefficients of the chroma color channels in MB to original values (meaning that macroblock MB is *non-data-embeddable*).
5. Repeat Step 4 until the entire secret message in array A has been embedded (i.e., until all elements in A have been processed).

5.3 Recovery of Privacy-sensitive Contents

In this section, the proposed method for recovery of privacy-sensitive contents is introduced. In Section 5.3.1, the proposed idea is described, and the process for extraction and recovery privacy-sensitive contents is described in Section 5.3.2.

5.3.1 Proposed Idea

We use a secret key to extract the recovery information from an input privacy protected video. Once the privacy-sensitive contents in the input privacy protected video need to be recovered, the extracted recovery information is used to recover the original privacy-sensitive contents. According to the correct recovery information, we know the positions of the protected region in the input privacy protected video and the original privacy-sensitive contents.

5.3.2 Process for Extraction and Recovery Privacy-sensitive Contents

There are two phases in the proposed process for recovery of privacy information in a privacy protected video. The first is to extract the recovery information of the replaced region in the privacy protected video. The second is to replace the recovery information of the replaced region. The detailed algorithm is described in the following.

Algorithm 5.3 Process for extraction of the privacy-sensitive contents of a WebM privacy video.

Input: a WebM privacy video V ; and a secret key K and a random number generator f

used in Algorithm 5.2.

Output: a report E of the privacy-sensitive contents of the replaced region in V .

Steps.

1. Use the secret key K as a seed to generate a sequence Q of random numbers using the random number generator f ; and take the first eight random numbers from Q , denoted as N_1 through N_8 .
2. For each macroblock MB in the prediction frames of V , if the ROI map index in MB is set as 1, perform the following steps.
 - 2.1 Check the coefficients of each of the eight corresponding subblocks of the chroma color channels (four in the U channel and the other four in the V channel), denoted as SB_j , $j = 1, 2, \dots, 8$: if the coefficients of SB_j match those of data pattern i , then record the code of *data pattern i* as an *extracted result DP_i* .
 - 2.2 Combine DP_i with N_i by exclusive-OR operations to become a new data pattern P_i :
$$P_i = DP_i \oplus N_i \quad (5.5)$$
where $i = 1, 2, \dots, 8$.
 - 2.3 Transform P_i ($i = 1, 2, \dots, 8$) into a 4-bit binary string S_i and concatenate them to form a binary string $S_r = S_1S_2S_3S_4S_5S_6S_7S_8$. (e.g., if P_i is data pattern 3, then $S_i = 0011_2$)
3. Concatenate the entire S_r obtained in Step 2 to form a binary string B_r .
4. (*Reorganizing the message*) Use the secret key K as a seed to generate a sequence Q_r of random numbers using the random number generator f , de-randomize B_r with Q_r to get another binary string B_r' , and transform B_r' into a character form for use as the desired output report E .

Algorithm 5.4 Process for recovering the privacy-sensitive contents of a WebM privacy video.

Input: a WebM privacy video V , and a report E of the privacy-sensitive contents of the replaced region in V .

Output: a WebM video V' with recovered privacy-sensitive contents.

Steps.

Stage 1 --- initialize.

1. Set the recovery flag f to be 1.

Stage 2 --- process for recovering privacy-sensitive contents in the input video.

2. Perform the following steps for each prediction frame F in V .
 - 2.1 If f is set as 1, extract a set of unused recovery information from E and set f to be 0, where the recovery information includes the index of frame number F_r , the index of macroblock number M_r , the motion vector MV_r , and the frequency coefficients FC_r .
 - 2.2 Perform the following steps for each macroblock MB in F , assuming that the index of current frame F and the index of MB are denoted by F_r and M_r .
 - (1) Replace the motion vectors in MB to MV_r .
 - (2) Replace the corresponding frequency coefficients in MB to FC_r .
 - (3) Set f to be 1 and repeat Step 2.1, which means that the recovery information have been used.
 - 2.3 Repeat Steps 2.2 until f is set as 1 or until reaching the end of F .
3. Repeat Step 2 until reaching the end of V .

In Step 2.1, if a wrong key was used in Algorithm 5.3 to extract the hidden data,

the extracted recovery information in E will be just noise, cannot be used to recover the privacy-sensitive contents.

5.4 Experimental Results

Four surveillance videos are used in this experiment. The first one is a surveillance video which was acquired by a camera monitoring the aisle of Engineering Building 5 in National Chiao Tung University. Six representative frames of an original video are illustrated in Figure 5.5. In this surveillance video, we want to monitor activities around the aisle, but we hope that the personal information within the window of the second floor will not be revealed. Therefore, we utilized the proposed process for removing privacy-sensitive contents to remove the personal information. Six representative frames of a privacy protected video yielded by the proposed method for removing privacy-sensitive contents are shown in Figure 5.6. Six representative frames of a recovered video yielded by the proposed method for recovering privacy-sensitive contents are shown in Figure 5.7. Comparison between the original image and the corresponding recovered image is illustrated in Figure 5.17. The average value of PSNR of the recovered area with respect to the original protected area is 35.73 in Figure 5.17.

Another surveillance video used in our experiments came from monitoring the Computer Vision Lab at National Chiao Tung University and an aisle of the Library at National Chiao Tung University. Six representative frames of the original video are illustrated in Figure 5.8, Figure 5.11, and Figure 5.14. Six representative frames of a privacy protected video yielded by the proposed method of removing privacy-sensitive contents are shown in Figure 5.9, Figure 5.12, and Figure 5.15. Six representative frames of a recovered video yielded by the proposed method for

recovering privacy-sensitive contents are shown in Figure 5.10, Figure 5.13, and Figure 5.16. Comparison between an original image and the corresponding recovered image are illustrated in Figure 5.18, Figure 5.19, and Figure 5.20. The average values of PSNR of the recovered areas with respect to the original protected areas are 30.372 in Figure 5.18, 33.361 in Figure 5.19, and 31.374 in Figure 5.20.

5.5 Discussions and Summary

In this chapter, we have proposed a method for removing the privacy-sensitive contents in a selected privacy area in a WebM surveillance video, and hiding them into the video. Based on using the golden reference frame, we can modify the motion vectors and the frequency coefficients to remove privacy-sensitive contents, and embed them into the surveillance video for recovery use. We have also proposed a method for recovering privacy-sensitive contents in a privacy video. The privacy-sensitive contents hidden in the privacy video can be extracted to recover the original privacy-sensitive contents. Since some coefficients are not embedded into the privacy video for recovery use, some details of the recovered area may be lost. To solve this problem, we can embed all coefficients into the privacy video, but it may cause an increase of the size of the hidden data. This problem will be discussed in Chapter 6.

With this proposed privacy protection method, we can hide the privacy-sensitive contents in the surveillance video to avoid legal disputes and to protect the personal privacy of non-suspicious people. Moreover, the traditional object-based privacy protection method needs to recognize protected persons manually. By the proposed region-based privacy protection method, we can instead extract authorized persons automatically if they are in movement.



(a)

(b)



(c)

(d)



(e)

(f)

Figure 5.5 Six representative frames of a video. (a) The 41 frame. (b) The 53 frame. (c) The 77 frame. (d) The 99 frame. (e) The 124 frame. (f) The 239 frame.

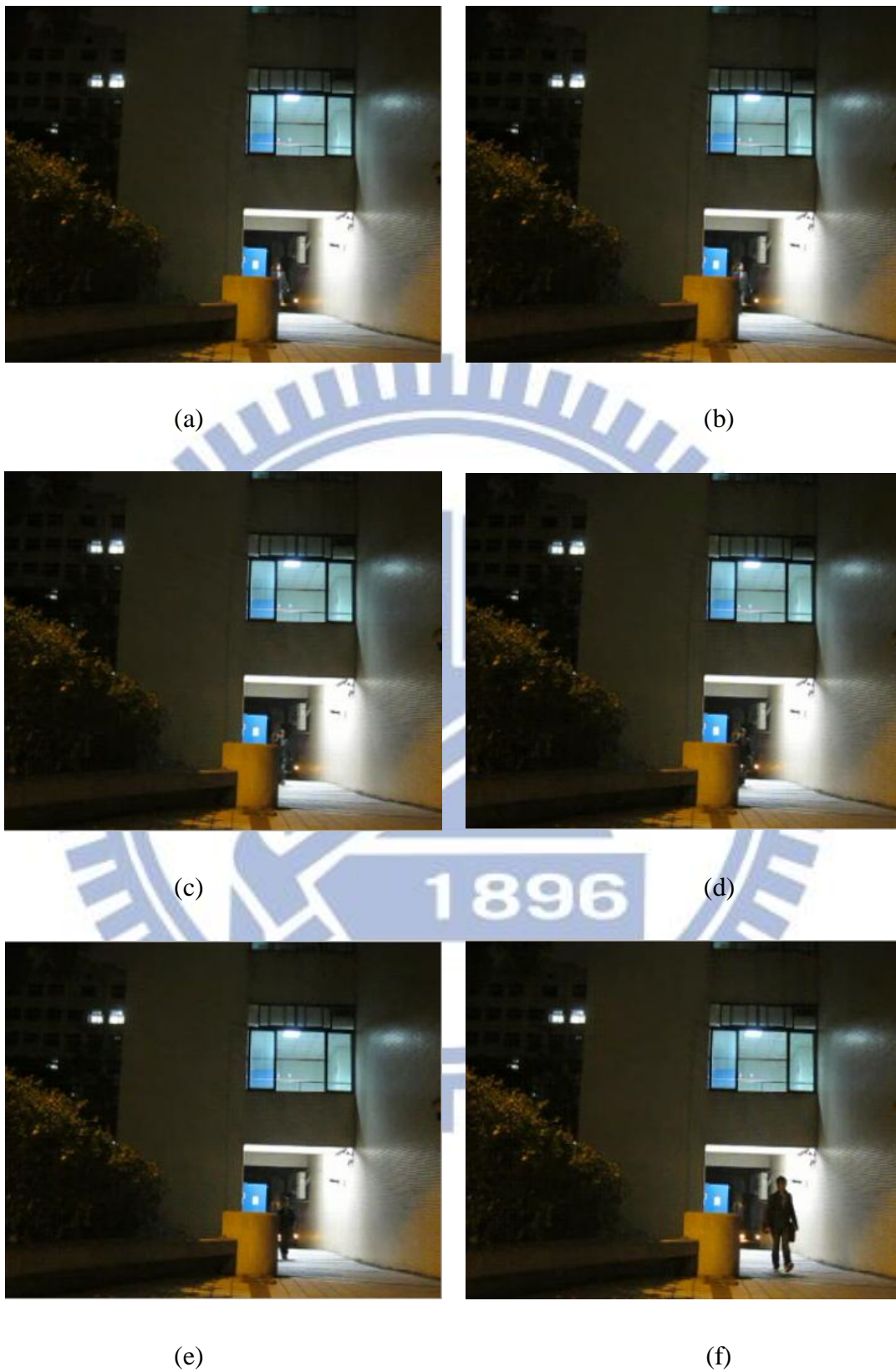


Figure 5.6 Six representative frames of the protected video of Fig. 5.5. (a) The 41 frame. (b) The 53 frame. (c) The 77 frame. (d) The 99 frame. (e) The 124 frame. (f) The 239 frame.

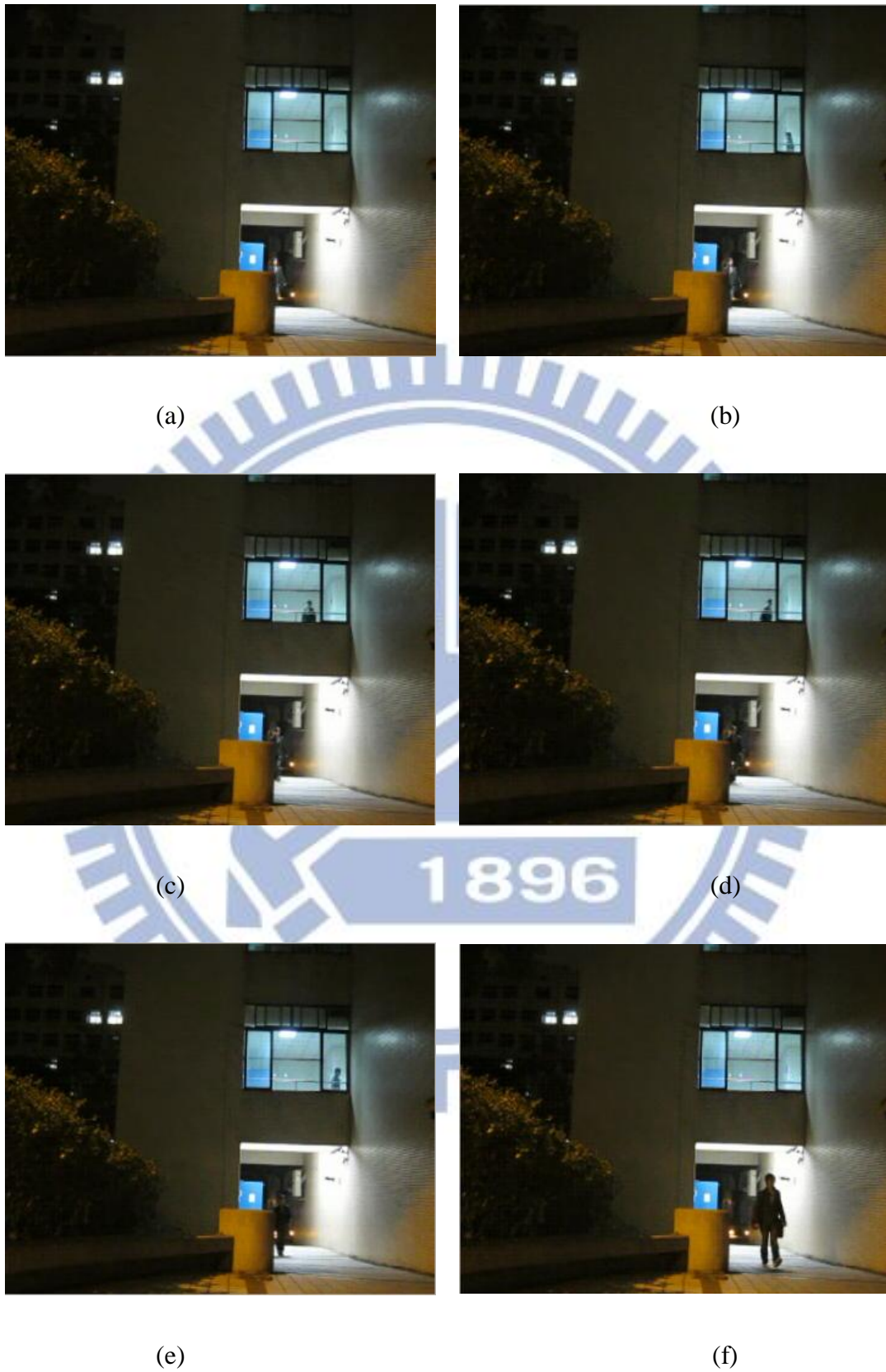


Figure 5.7 Six representative frames of a recovered video. (a) The 41 frame. (b) The 53 frame. (c) The 77 frame. (d) The 99 frame. (e) The 124 frame. (f) The 239 frame.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 5.8 Six frames of a second video. (a) The 40th frame (b) The 41th frame (c) The 42th frame (d) The 43th frame (e) The 44th frame (f) The 45th frame.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 5.9 Six frames of the protected video of Fig. 5.8. (a) The 40th frame (b) The 41th frame (c) The 42th frame (d) The 43th frame (e) The 44th frame (f) The 45th frame.



(a)



(b)



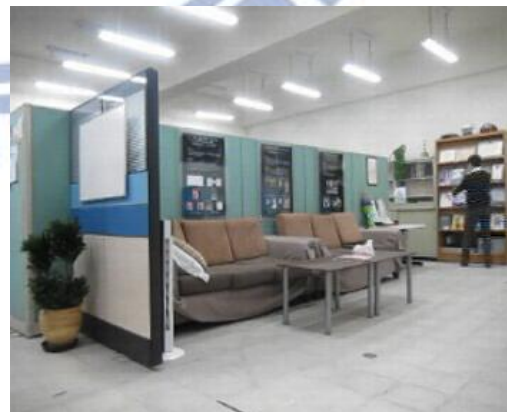
(c)



(d)



(e)



(f)

Figure 5.10 Six frames of the recovered video. (a) The 40th frame (b) The 41th frame (c) The 42th frame (d) The 43th frame (e) The 44th frame (f) The 45th frame.



(a)

(b)



(c)

(d)



(e)

(f)

Figure 5.11 Six frames of a third video. (a) The 131th frame (b) The 137th frame (c) The 147th frame (d) The 160th fame (e) The 174th frame (f) The 181th frame.



(a)

(b)



(c)

(d)



(e)

(f)

Figure 5.12 Six frames of the protected video of Fig. 5.11. (a) The 131th frame (b) The 137th frame (c) The 147th frame (d) The 160th frame (e) The 174th frame (f) The 181th frame.



(a)

(b)



(c)

(d)



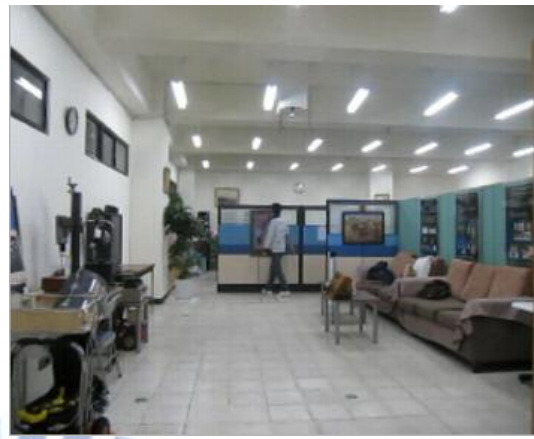
(e)

(f)

Figure 5.13 Six frames of the recovered video. (a) The 131th frame (b) The 137th frame (c) The 147th frame (d) The 160th frame (e) The 174th frame (f) The 181th frame.



(a)



(b)



(c)



(d)

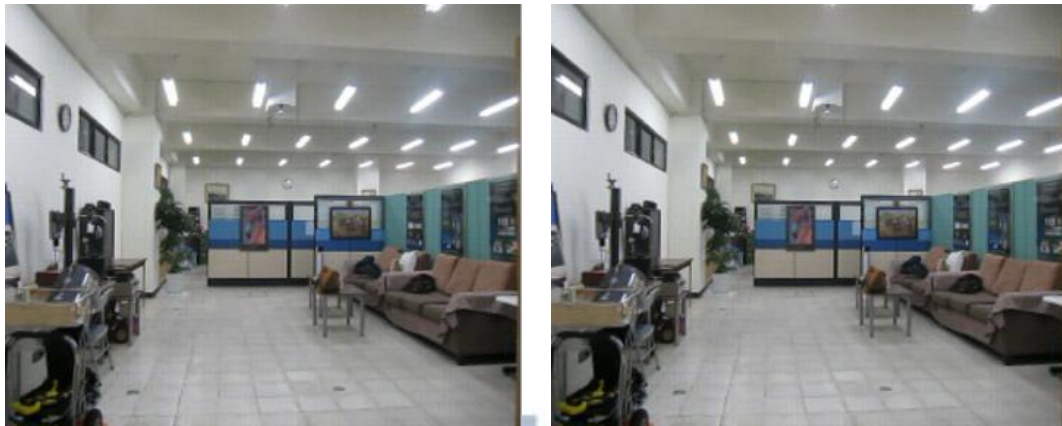


(e)



(f)

Figure 5.14 Six frames of a fourth video. (a) The 195th frame (b) The 209th frame (c) The 223th frame (d) The 236th frame (e) The 246th frame (f) The 255th frame.



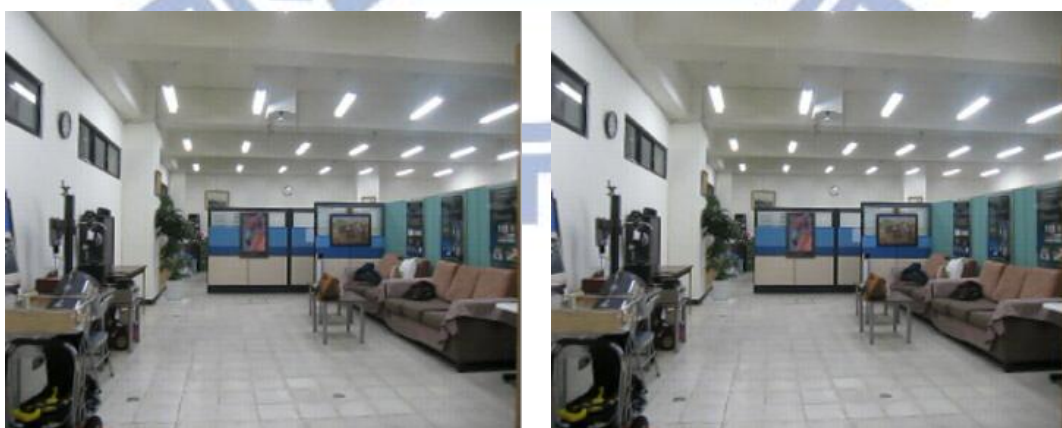
(a)

(b)



(c)

(d)



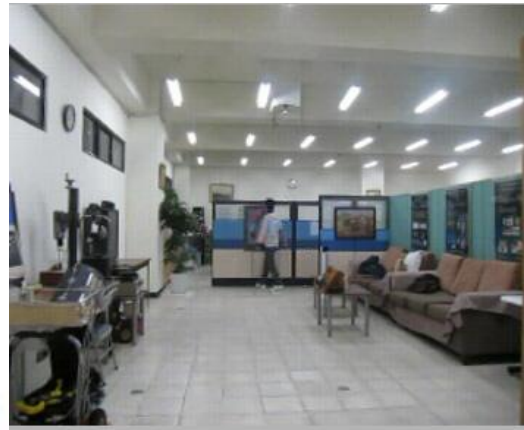
(e)

(f)

Figure 5.15 Six frames of the protected video of Fig. 5.14. (a) The 195th frame (b) The 209th frame (c) The 223th frame (d) The 236th frame (e) The 246th frame (f) The 255th frame.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 5.16 Six frames of the recovered video. (a) The 195th frame (b) The 209th frame (c) The 223th frame (d) The 236th frame (e) The 246th frame (f) The 255th frame.



(a)

(b)

Figure 5.17 Comparison between an original image and the corresponding recovered image. (The 77th frame) The average value of PSNR of the recovered area with respect to between the original protected area is 35.73. (a) The original image. (b) The recovered image.



(a)

(b)

Figure 5.18 Comparison between an original image and the corresponding recovered image. (The 43th frame) The average value of PSNR of the recovered area with respect to between the original protected area is 30.372. (a) The original image. (b) The recovered image.



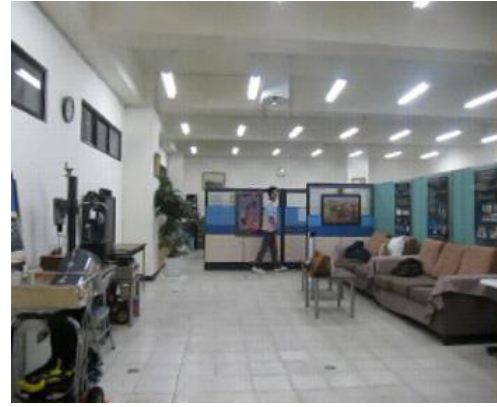
(a)

(b)

Figure 5.19 Comparison between an original image and the corresponding recovered image. (The 147th frame) The average value of PSNR of the recovered area with respect to between the original protected area is 33.361. (a) The original image. (b) The recovered image.

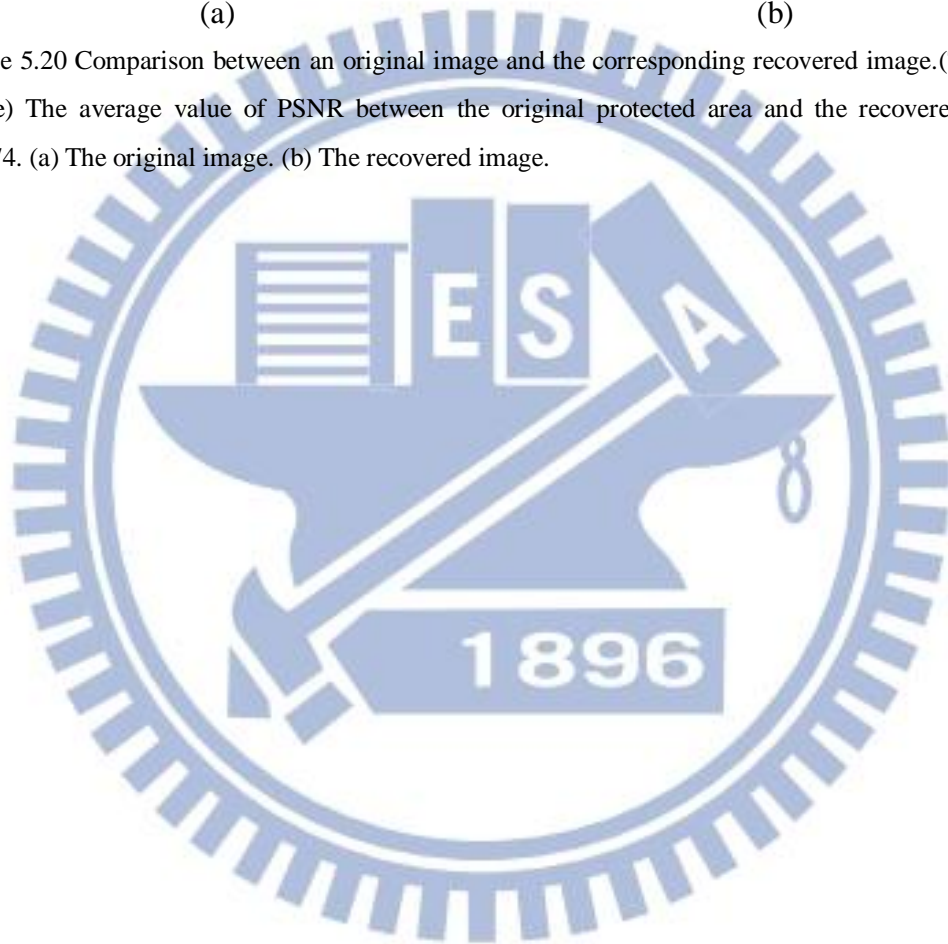


(a)



(b)

Figure 5.20 Comparison between an original image and the corresponding recovered image. (The 246th frame) The average value of PSNR between the original protected area and the recovered area is 31.374. (a) The original image. (b) The recovered image.



Chapter 6

Conclusions and Suggestions for Future Works

6.1 Conclusions

In this study, we have proposed several methods for a variety of information hiding applications via WebM videos, such as covert communication, authentication, privacy protection, etc.

For covert communication, a method for transmitting secret information via WebM videos has been proposed. The proposed method modifies the frequency coefficients of the chroma color channels in the compression result and generates encoded data patterns for data hiding. In addition, the PSNR values are computed and compared with a threshold to optimize these changes for maintaining the video quality. For secret security enhancement, a scheme has been proposed as well to select randomly positions for message data hiding in images, preventing malicious users from figuring out the data embedding locations and carrying out attacks.

For video authentication, a motion detection scheme using prediction modes and motion vectors to detect motion regions in an input video has been proposed. And a method for authentication of surveillance videos by hiding motion-region information has been proposed accordingly. Motion-region information and frame indexes of an input video are used to generate authentication signals. The authentication signals can be utilized to detect and to verify tampering in a suspicious video.

For privacy protection in surveillance videos, a method using WebM video

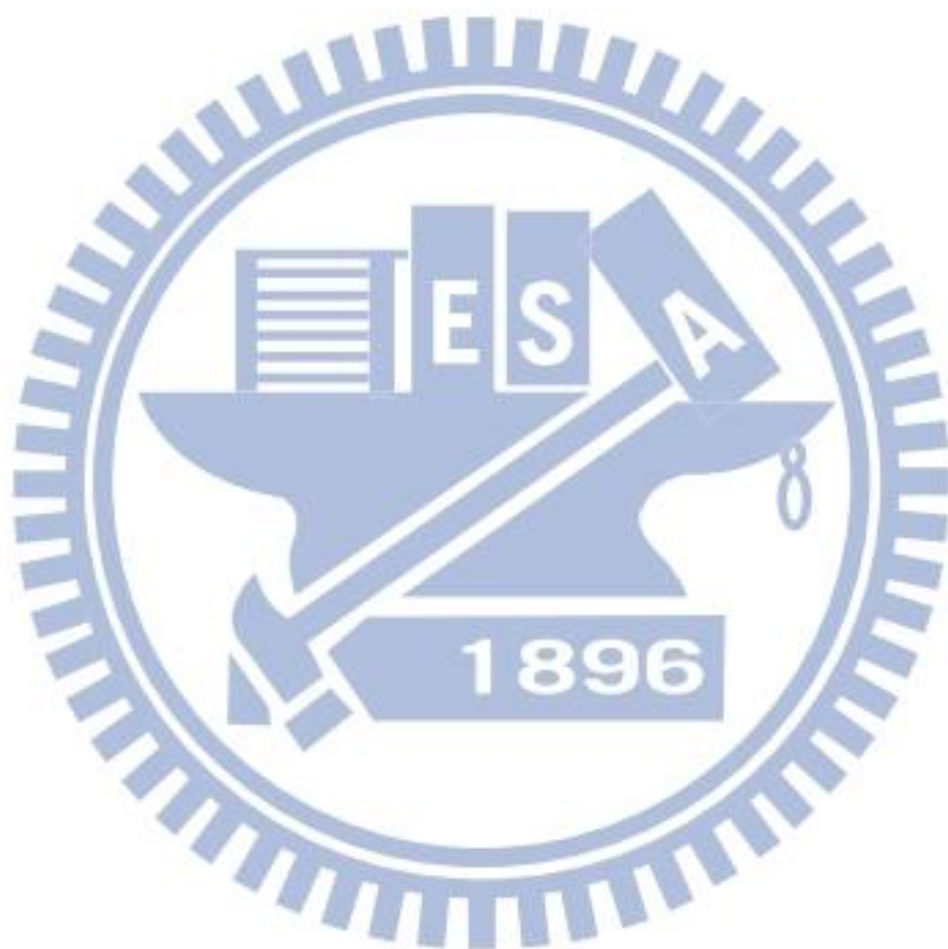
features for removing privacy information in a WebM surveillance video has been proposed. An authorized user can specify a protected region in an input video, and the privacy-sensitive contents of the protected region can be removed to protect the privacy-sensitive contents within the protected region. The removed privacy-sensitive contents is not eliminated but embedded into the video in case there is a need of retrieving the privacy-sensitive contents.

Good experimental results of the proposed methods show the feasibility for real applications of covert communication, video authentication, and privacy protection.

6.2 Suggestions for Future Works

Several suggestions for future works are listed in the following.

1. The features used to detect video contents can be extended to include more types, such as color information, user characteristics, etc.
2. The way to modify DCT coefficients in the proposed data hiding method for WebM videos can be modified to reduce distortion more effectively.
3. The decoding information used to recover privacy information can be expanded to include all the luma color channel coefficients in order to recover more details of the privacy information.
4. It is interesting to extend these video applications to handle videos of other video standards, such as H.265.
5. It is also interesting to extend the data hiding method proposed in this study to integrate other data hiding methods in order to increase the hiding capacity.
6. It is interesting to extend the proposed authentication method to handle a frame without motion objects or with motion objects that are added by malicious users.



References

- [1] Y. Hu, et al., "Information hiding based on intra prediction modes for H.264/AVC," *Proceedings of IEEE International Conference on Multimedia and Expo*, Beijing, China, pp. 1231-1234, Jul., 2007.
- [2] Hussein A. Aly, "Data Hiding in Motion Vectors of Compressed Video Based on Their Associated Prediction Error," *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 14-18, March, 2011.
- [3] M. Yang and N. Bourbakis, "A High Bitrate Information Hiding Algorithm for Digital Video Content under H.264/AVC Compression," *Proceedings of IEEE International Conference on Image Processing Midwest Symposium on Circuits and Systems*, Cincinnati, OH, USA, vol. 2, pp. 935- 938, Aug., 2005.
- [4] S. K. Kapotas, et al., "Data hiding in H.264 encoded video sequences," *Proceedings of International Workshop on Multimedia Signal Processing*, Chania, Crete, Greece, pp. 373-376, Oct., 2007.
- [5] I. Haritaoglu, D. Harwood, and L. S. Davis, "W⁴: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 809-830, Aug. 2000.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. Zhang and A. T. S. Ho, "Efficient video authentication for H.264/AVC," *Proceedings of the First International Conference on Innovative Computing, Information and Control*, Beijing, China, vol. 3, pp. 46-49, Aug. 2006.
- [11] D. Pröfrock, H. Richter, M. Schlawweg, and E. Müller, "H.264/AVC video authentication using skipped macroblocks for an erasable watermark," *Proceedings of Visual Communication and Image Processing*, Beijing, China, vol. 5960, pp. 1480-1489, Jul. 2005.
- [12] K. Saadi, A. Bouridane, and A. Guessoum, "H.264/AVC video authentication based video content", *Proceedings of International Symposium on I/V Communications and Mobile Network*, Las Vegas, Nevada, USA, pp. 1-4, December 2010.
- [13] F. Dufaux, T. Ebrahimi, and S. A. Emitall, "Smart video Surveillance System Preserving Privacy," *Proceedings of SPIE Image and Video Communications and Processing*, San Jose, CA, USA, vol. 5685, pp. 54-63, Jan. 2005.
- [14] P. Meuel, M. Chaumont, and W. Puech "Data Hiding in H. 264 Video for Lossless Reconstruction of Region of Interest," *Proceedings of European Signal Processing Conference*, Poznań, Poland, pp. 120-124, Sept. 2007.
- [15] W. Zhang, S.-C. S. Cheung, and M. Chen, "Hiding privacy information in video surveillance system," *IEEE International Conference on Image Processing*, Genova, Italy, vol. 3, pp. 868-871, Sept. 2005.
- [16] X. Yu, K. Chinomi, T. Koshimizu, N. Nitta, Y. Ito, and N. Babaguchi, "Privacy

protecting visual processing for secure video surveillance,” *Proceedings of IEEE International Conference on Image Processing*, Los Alamitos, CA, USA, pp. 1672-1675, Oct. 2008.

[17] K. John. (2010). *The WebM project*. [Online]. Available:

<http://www.webmproject.org/>

[18] S. Winkler, C. J. van den Branden Lambrecht, and M. Kunt (2001). *Vision and Video: Models and Applications* Springer, USA, 2001.

[19] A.V. Oppenheim and R. W. Schaffer. (1991). *Discrete-time signal processing*. (2nd ed.), Prentice Hall, USA, 1991.

[20] J. Lie and K. Ngi Ngan, “An Error Sensitivity-based Redundant Macroblock Strategy for Robust Wireless Video Transmission,” *Proceedings of IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, Mawii, Hawaii, USA, pp. 1118-1123, Sept. 2005.

