Document Title:     Development and Qualitative Evaluation of Steganalysis and Digital Forgery Systems

Author(s):          Qingzhong Liu, Ph.D., Peter A. Cooper, Ph.D., Andrew H. Sung, Ph.D.

Document Number:    311502

Date Received:      February 2026

Award Number:       2010-DN-BX-K223

**Final Report:**

**Development and Qualitative Evaluation of Steganalysis and Digital Forgery Systems**

**Award Number:**

**2010-DN-BX-K223**

**Authors:**

**Qingzhong Liu, Ph.D., Principal Investigator**
**Department of Computer Science**
**Sam Houston State University**
**Huntsville, TX 77341**
**Phone: 936 294 3569**
**Email: liu@shsu.edu**

**Peter A. Cooper, Ph.D., Co- Principal Investigator**
**Department of Computer Science**
**Sam Houston State University**
**Huntsville, TX 77341**
**Phone: 936 294 1569**
**Email: cooper@shsu.edu**

**And**

**Andrew H. Sung, Ph.D., Co- Principal Investigator**
**Department of Computer Science and Engineering, and Institute for Complex Additive Systems Analysis**
**New Mexico Institute of Mining and Technology**
**Socorro, NM 87801**
**Phone: 505 835 5949**
**Email: sung@cs.nmt.edu**

# Abstract

Steganography, the ancient art for secretive communications, has revived on the Internet by hiding secret data in completely imperceptible manners, and has created a serious threat due to the covert channel that can be readily exploited for various illegal purposes. Likewise, multimedia tampering, which has been greatly facilitated and proliferated by various multimedia processing tools, is increasingly causing problems concerning the authenticity of digital multimedia data. There is a critical need to develop reliable methods for steganography detection or steganalysis and for forgery detection to serve purposes in national security, law enforcement, and cybercrime fighting.

To detect steganography and forgery on multimedia data, our research goals include discovering the characteristic modification caused by digital multimedia steganography and forgery, developing more accurate and more reliable methods for steganalysis and digital evidence authentication, and developing a complete evaluation procedure for gaining full understanding of the accuracy, reliability, and measurement validity of steganography detection and digital evidence authentication in digital image, audio, and video files.

To achieve these goals, the procedures of our research design and methods are conducted as follows:

1. Construct a comprehensive and high volume multimedia steganography and forensic database.

2. Analyze the bias and variation of each confirmed source by using existing methods; and by developing new methods, improve the quantification of the characteristics and uncertainties of the cover, steganography, and forgery, created by these sources, and provide a more complete evaluation in different circumstances including multimedia type and format, signal complexity, source type, information-hiding/forgery type and modified size, detection method, detection accuracy, the strength and limitation of a certain method in which circumstance.

3. Measure detection performance.

4. Monitor and improve the steps in the forensic evidence analysis process in digital media by integrating updated methods with the use of data mining and computational intelligence techniques for steganography detection and forgery detection.

We conjecture that data hiding in steganography and manipulation in forgery production change the statistics of original multimedia data, and hence leave the clues of modification. Our study aims to discover the features that may discriminate the manipulations from intactness and analyze different patterns caused by different operations. In this project, we have developed several novel detection algorithms based on feature mining and machine intelligence techniques in detecting steganography, forgery manipulation and relevant operations such as cropping, double compression on multimedia data. Our experimental results validate our hypothesis and indicate that our

methods have obtained the detection performances in detecting several types of steganography and forgery on multimedia data within the state-of-the-art. Our study also shows that a complete evaluation of the detection performance of different algorithms should include image/signal complexity–in addition to other relevant factors such as hiding ratio or compression ratio–as a significant and independent parameter for some detections including JPEG double compression.

# Table of Contents

# Acknowledgements

# Executive Summary

**Synopsis of the Problem**

Steganography, the art and science of carrying messages in covert channels, and forgery have revived in digital realms. The detection of steganography and forgery on multimedia data has important impacts on public safety and national security.

Several steganographic algorithms/systems have been proposed, including LSB embedding, LSB matching (Mielikainen 2006), spread spectrum steganography (Marvel, Boncelet and Retter 1999), Outguess (Provos 2001), F5 (Westfeld 2001), model-based steganography (Sallee 2003 and 2005), Steghide (Hetzl and Mutzel 2005), BCH syndrome code based less detectable JPEG steganography (Sachnev, Kim and Zhang 2009), and highly undetectable steganography (HUGO) (Pevny, Filler and Bas 2010). Although these steganographic systems have been successfully steganalyzed (Chen and Shi 2008; Fridrich 2004; Fridrich, Kodovsky, Holub and Goljan 2011a, 2011b; Fu and Shi et al. 2006; Gu and Kurugollu 2011; Ker 2004; Kharrazi, Sencar and Memon 2006; Kodovsky and Fridrich 2009, 2011, 2012; Kodovsky, Pevny and Fridrich 2010; Kodovsky, Fridrich and Holub 2012; Li, Shi and Huang 2009; Liu, Sung and Ribeiro 2005; Liu et al. 2006; Liu Sung and Qiao 2008; Liu et al. 2008a, 2008b, 2008c; Liu and Sung 2007; Liu et al 2011a, 2011b, 2011c; Liu 2011a; Pevny and Fridrich 2007, 2008; Shi et al. 2007), the advances in steganography have posed new challenges to steganalyzers (Filler and Fridrich 2010; Filler, Judas and Fridrich 2011; Filler and Fridrich 2011; Solanki, Sarkar and Manjunath 2007).

The potential of exploiting steganography for covert dissemination is of increasing concern; a recent espionage case revealed that steganography had been employed by a foreign government intelligence agency (Web justice 1; Web justice 2). Secretbook (web secretbook), a Google Chrome extension allows users to transmit completely secret messages on Facebook, and the hidden message in photos cannot be scanned for keywords by Facebook or read by prying friends.

In multimedia forgery, double compression is an indispensable operation, and it is an effectual forensic indicator to recover the processing history. In digital multimedia, JPEG (an acronym for the Joint Photographic Experts Group, which created the image compression standard) and MPEG (an acronym for the Moving Picture Experts Group, formed by the International Organization for Standardization and the International Electro-technical Commission to set standards for audio and video compression and transmission) are the most popular lossy compression standards. Today's digital techniques make it easy to tamper JPEG images and MPEG files without leaving any visible clues; since most tampering involves JPEG/MPEG double compression (the original JPEG/MEPG files are manipulated in spatial/temporal domain and then saved in JPEG/MPEG files), it heightens the need for accurate analysis of JPEG/MPEG double compression in image forensics.

In image forensics, while most methods target traditional image tampering, seam carving-

based image tampering in JPEG format has been ignored to some extent. Seam carving, an algorithm for image resizing, is known as content-aware scaling, liquid resizing or liquid rescaling. It allows the removal of selected whole objects from photographs. The seam carving method for content-aware resizing and object removal has been implemented in Adobe Photoshop CS4 (Web photshop-cs4), GIMP (Web liquidrescale), digiKam (Web digikam), ImageMagick (Web imagemagick), as well as stand-alone programs such as iResizer (Web iresizer). The proliferation of seam-carved images presents a challenge to authorities who require image authentication.

In audio forensics, a few of algorithms have been presented to detect the forgery or related manipulation in audio streams, including the detection of double compression on MPEG-1 Audio Layer 3 or MP3 audio streams (Yang et al. 2008; Qiao, Sung and Liu 2010). Advanced Audio Coding (AAC), a lossy audio compression scheme, standardized by the International Organization for Standardization and the International Electro-technical Commission, which was designed to be the successor of the MP3 format, generally obtains better sound quality than MP3 at similar bit rates. While AAC audio files widely spread (Web AAC), to our knowledge, the literature of the forgery detection of AAC audio files is still missing to this date.

**Purpose**

To detect steganography and forgery on multimedia data, our research goals include:

1) Discovering the characteristic modification caused by digital multimedia steganography and forgery;

2) Developing more accurate and more reliable methods for steganalysis and digital evidence authentication;

3) Developing a complete evaluation procedure for gaining full understanding of the accuracy, reliability, and measurement validity of steganography detection and digital evidence authentication in digital image, audio, and video files.

**Research Design**

Our research design is conducted in the following procedures:

1) Construct a comprehensive and high volume multimedia steganography and forensic database.

2) Analyze the bias and variation of each confirmed source by using existing methods; and by developing new methods, improve the quantification of the characteristics and uncertainties of the cover, steganography, and forgery, created by these sources, and provide a more complete evaluation in different circumstances including multimedia type and format, signal complexity, source type, information-hiding/forgery type and modified size, detection method, detection accuracy, the strength and limitation of a certain method in which circumstance.

3) Measure detection performance.

4) Monitor and improve the steps in the forensic evidence analysis process in digital media by integrating updated methods with the use of data mining and computational intelligence techniques for steganography detection and forgery detection.

We surmise that steganography and forgery manipulation will alter some features of original multimedia data, and hence leave the clues of being touched. Our study aims to discover these features. We have developed several novel detection algorithms based on feature mining and machine intelligence techniques in detecting steganography, forgery manipulation and relevant operations.

We have designed several types of features including neighboring joint density on quantized discrete cosine transform (DCT) coefficients in JPEG images and MP3 audio files that discriminate the steganograms, doctored image/audio files from the untouched. The detection of MPEG double compression is also in-depth investigated. In our study, a shift-recompression-based framework is proposed with new feature sets to detect steganography and forgery in JPEG images and MP3 and AAC audio streams.

In this study, in addition to our previously proposed neighboring joint density-based approach (Liu, Sung and Qiao 2011a), we designed several novel approaches including calibrated neighboring joint density-based approaches to expose the manipulation to original JPEG files.

With regard to AAC audio forgery, from our standpoint, similar to JPEG compression AAC audio compression introduces block (frame) artifacts in tampering, and accordingly, we propose a shift-recompression-based differential analysis to detect the forgery in AAC audio streams with the same compression bit rate.

**Key Findings**

1. Most steganography and forgery manipulation change the statistics of original multimedia data.

2. Neighboring joint density and other statistical features are effective to detect several types of steganography and forgery manipulations. Neighboring joint density-based feature mining under different shift recompression have gained the highest detection accuracy in detecting several types of steganographic systems, and delivered the state-of-the-art detection results in JPEG-based steganalysis.

3. By combining neighboring joint density with spatial domain-based rich models that was designed for steganalysis, we have noticeably improved the detection of seam-carved forgery in JPEG images that was re-encoded at the same compression quality after doctoring.

4.  Shift-recompression-based SRSC feature set is effective to detect AAC audio forgery that was encoded at the same compression quality. To our knowledge, such kind of detection has not been literally exposed.

5.  Double compression in JPEG images and MEPG video files have modified several statistics in DCT domain. Normally it is much easier and much reliable to detect the double compression with the last compression quality is higher than the first compression quality. However, it is not so easy to detect the double compression complying with the second compression is much heavier than the first compression.

6.  The detection evaluation may be more meaningful and complete by considering the data format, payload hiding ratio, hiding algorithms and hiding parameters, detection methods, and image/signal complexity. In the detection of the same double compression at different image complexity, the image complexity is higher, then the detection accuracy is lower.

7.  It is still hard to break some types of steganographic systems. For example, we have designed an statistically invisible steganography based on the complexity of the DCT blocks in JPEG images, and improved an existing steganographic algorithm, YASS (acronym for Yet Another Steganographic Scheme, Solanki, Sarkar and Manjunath 2007), to implement completely randomized embedding  in DCT blocks. The detection of such steganographic systems is still very difficult wherein the hiding parameters are meticulously selected or the hiding algorithms are specially designed against the detection.

**Conclusions**

*Discussions of Key Findings*

1.  In detecting JPEG-based steganography and forgery, we analyze the neighboring joint density of the DCT coefficients and reveal the difference between an untouched image and the modified version by steganography and forgery manipulation. In real detection, untouched image and the modified version may not be obtained at the same time, and different JPEG images may have different neighboring joint density features. To produce the self-calibration, we design the reference features of neighboring joint density features under different shift recompression, and propose calibrated neighboring joint density-based approaches to distinguish steganograms and altered images from untouched ones. Our study shows that this approach has multiple promising applications in image forensics. Compared to the state-of-the-art of steganalysis detectors, our approaches deliver better or comparable detection performances with a much smaller feature set to detect several steganographic systems including DCT-embedding-based adaptive steganography and YASS. Our method is also effective to detect seam-carved forgery in JPEG images. By integrating calibrated neighboring density with spatial domain rich models that were originally designed for steganalysis, the hybrid approach obtains the best detection accuracy to discriminate

seam-carved forgery from an untouched image in JPEG format. Our study shows that it is a promising manner by exploring steganalysis and forgery detection together.

It is still hard to break YASS while the steganograms are produced by using a small *noused* parameter. In detecting seam carving forgery, rich models provide a marked improvement with abundant features.

2. In detecting JPEG double compression, we have developed a technique that can successfully detect JPEG double compression by integrating marginal density and the neighboring joint density features in DCT domain. Our method is superior to Markov process-based approach in terms of achieving a higher detection accuracy at a lesser computational cost. Our study shows that the detection performance is related not only to the compression quality factors but also to image complexity, which is an important parameter that seems to have been so far overlooked by the research community in conducting performance evaluation. To formally study the performance evaluation issues, the image complexity and compression quality should therefore be included as a whole.

Following the success in detection of JPEG double compression, we conduct studies based on processed smartphone images to identify the smartphone source and the post-capture manipulations. Experimental results show that our method is strongly promising in correctly identifying the smartphone source and revealing the past manipulations simultaneously, including the combination of double JPEG compression, cropping, and rescale. Our studies also indicate that, due to the complexity of intentional manipulation, it is more productive to combine clustering and classification techniques together for performing the detection.

3. In detecting double MPEG compression, we offer a qualitative statistical analysis about the impact caused by MPEG-2 compression on distributions of reconstructed DCT coefficients, and demonstrate the differences in distributions of quantized DCT coefficients between the single compression and double compression. A set of DCT distributions with different quantization scale factors are constructed to extract convex pattern features, and a novel detection algorithm is designed to detection of double MPEG compression in CBR (constant bitrate) videos. In our simulation system, the target output bit-rate, rather than quantization scale factor, is selected as the only parameter to control MPEG-2 encoders. The target output bit-rate can easily be configured through the system menu, without need to modify source codes of MPEG-2 encoders. So it makes our detection algorithm more suitable for all kinds of video coding systems, especially in some business video systems. On the other hand, our proposed detection algorithm maintains good detection performance in many cases. More specifically, it can detect double compressed videos with both high-quality and low-quality. Even if the primary compression and the secondary compression use different kinds of MPEG-2 encoders, our algorithm can also reveal the track of double MPEG-2 compression.

4. In detecting MP3-based audio steganography, we design a detection method by extracting frequency-based sub-band moment statistics as well as accumulative

neighboring joint probabilities and accumulative Markov transition probabilities in the compression domain. Generalized Gaussian density (GGD) is introduced to estimate the distribution of the modified discrete cosine transform (MDCT) coefficients. We also propose moment statistics of GGD shape parameters ($\beta$) extracted from individual frames as features, and utilize the shape parameter from the whole audio clip as a measure of signal complexity. The relation between audio steganalysis performance and signal complexity is also studied experimentally. Three feature selection methods are employed to further enhance the detection accuracy. Our approach leads to a successful detection of information-hiding in MP3 audio, under each category of signal complexity and modification density, especially in audio with a high signal complexity and a low modification density (Qiao, Sung and Liu 2013).

5. In detecting AAC audio forgery, we propose a shift-recompression-based SRSC feature mining with machine learning techniques to reveal the difference between untouched AAC audio streams and doctored AAC audio streams that was re-encoded at the same bit rate after tampering. Our experimental results show that our approach is very promising and effective.

6. To conclude, by exploring and developing new detection algorithms/approaches and new measurement parameters in multimedia forensics, we have successfully achieved the project goals including "discovering the characteristic modification caused by digital multimedia steganography and forgery, developing more accurate and more reliable methods for steganalysis and digital evidence authentication, and developing a complete evaluation procedure for gaining full understanding of the accuracy, reliability, and measurement validity of steganography detection and digital evidence authentication in digital image, audio, and video files."

*Implications for policy and practice*

Multimedia forensics is a multiple-disciplinary research field with important impacts to law enforcement. In multimedia forensics, steganography detection or steganalysis and forgery detection are two spots. Multimedia steganography and forgery techniques and the proliferation have made big challenges to law enforcement.

By exploring the characteristic modification caused by digital multimedia steganography and forgery, developing new detection algorithms/approaches, and adopting new measurement parameters for the evaluation, we have successfully achieved the project goals including "discovering the characteristic modification caused by digital multimedia steganography and forgery, developing more accurate and more reliable methods for steganalysis and digital evidence authentication, and developing a complete evaluation procedure for gaining full understanding of the accuracy, reliability, and measurement

validity of steganography detection and digital evidence authentication in digital image, audio, and video files."

The implications for policy and practice lie in the following:

1. Although multimedia steganography and forgery have made big challenges to law enforcement in protection of public safety and national security, our study shows that some advanced steganography and forgery systems can be accurately detected and hence the relevant crimes may be defeated and/or prevented. For example, in detecting several types of JPEG steganography at the relatively high information hiding ratio, our approach has achieved 100% mean accuracy over 100 experiments. Our forgery detection approaches in this study are also very promising with high detection accuracies. Therefore, we would recommend utilize the state-of-art for steganography and forgery detection for forensics purposes.

2. The complete evaluation in multimedia forensics may include multiple parameters including information hiding ratio and/or forgery size, compression factors, hiding algorithms, multimedia signal complexity, detection algorithms, feature selection methods, classification models and learning classifiers.

3. Our study also indicates that it is still hard to defeat some meticulously designed steganography, e.g., the data hiding takes place in the high complexity components in the multimedia signal (Liu Sung Chen and Huang 2011).

4. The study in multimedia forensics is normally subjected to relatively simple environment with a certain knowledge and limitation to the testing multimedia data. For example, to detect some type of steganography by using a steganographic algorithm x, the steganograms are denoted as Sx, covers are denoted as C. Classification models are constructed to discriminate the steganogram from cover. It is clear that the detection is conducted in the environment that contains only Sx and C, and the outcome can be predicted either Sx or C, relatively predestinated.

   Unfortunately, the real life detection generally faces an open and complicated environment. For example, we are given a JPEG image to determine whether it is carrying a covert message or not. We cannot simply adopt the classification model that was used to distinguish Sx and C, since we are not sure about the image under examination is either an untouched cover or the type Sx of steganogram, or some other type of steganogram, or a cover that was processed by some legitimate operations.

**5.** It is known that that steganography had been employed by a foreign government intelligence agency (Web justice 1; Web justice 2), and the potential usage of steganography to disseminate covert message in social media such as on Facebook could be enormous (web secretbook). The further study in multimedia forensics is highly needed for forensics purposes.

*Implications for further research*

The continuous improvement of the state of steganalysis and forgery detection should be emphasized in the future study. Additionally, several new steganographic systems have been proposed to hide data in JPEG images (Liu, Sung, et al. 2011; Holub and Fridrich 2013), AAC audio streams (Wei, Li and Wang 2010), and VoIP audio stream including Skype-based steganography (Mazurczyk, Karas and Szczypiorski 2013), and no effective detection methods area available to this date, which is worthy for the further exploration.

While we have designed several effective detection approaches within the state-of-the-art, the realistic detection toolkits may be implemented for the testing and validation for forensics purposes.

It is worthy of making the contribution for real life detection that generally faces an open and complicated environment. Further study may be also highlighted on revealing the processing history of the multimedia data under the examination.

# I.    Introduction

**Statement of the Problem**

Multimedia forensics is a multiple-discipliniary research field with important impacts on the protection of public safety and enhancement of national security. In multimedia forensics, steganography detection or steganalysis and forgery detection are two active areas and are generally separately studied, although both continue to face challenges.

Steganography, Greek for covered writing, is the art and science of carrying messages in covert channels, aiming to enable secretive communication by embedding data into digital files without attention to the existence of the hidden message. The potential of exploiting steganography for covert dissemination is of increasing concern; an espionage case revealed that steganography had been employed by a foreign government intelligence agency (Web justice 1; Web justice 2). Recently, a Google Chrome extension allows us to hide secrets in Facebook photos (Web secretbook).

Fake photos have employed for decades, and with various image processing tools, digital images can now be easily forged. Figure I-1 shows some examples of image forgery. Generally, tampering manipulation on a JPEG image involves several different basic operations, such as image rescaling, rotation, splicing, double compression, etc. While we decode the bit stream of a JPEG image and implement the manipulation in spatial domain, and then compress the modified image back to JPEG format, if the quantization matrices are different between the original JPEG image and the modified, the modified JPEG image has undergone a double JPEG compression. Although JPEG double compression does not by itself prove malicious or unlawful tampering, it is an evidence of image manipulation.
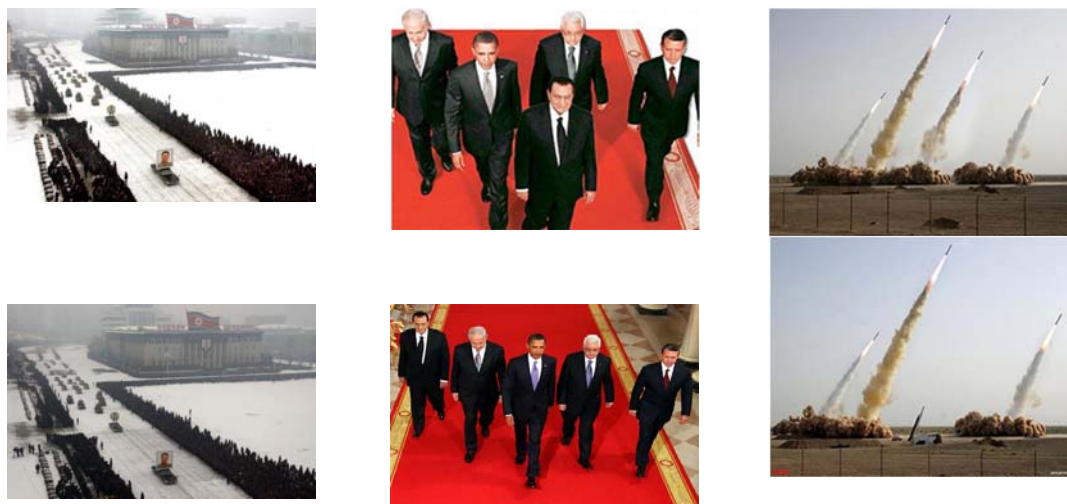


**Figure I-1**.  Image forgery examples (Web cbsnews1; Web cbsnews2; Web latimesblogs; Web theblaze), Tampered photos are shown on the upper and original ones are shown on the below.

Figure I-2 shows two source images in JPEG and a tampered image composited from the two sources. All three images are downloaded from *worth1000.com*. The quantization matrices affiliated with the luminance parts of these three JPEG images are given with different quantization values.



$$\begin{bmatrix} 11 & 7 & 7 & 11 & 16 & 26 & 34 & 40 \\ 8 & 8 & 9 & 13 & 17 & 38 & 40 & 36 \\ 9 & 9 & 11 & 16 & 26 & 38 & 46 & 37 \\ 9 & 11 & 15 & 19 & 34 & 57 & 53 & 41 \\ 12 & 15 & 24 & 37 & 45 & 72 & 68 & 51 \\ 16 & 23 & 36 & 42 & 53 & 69 & 75 & 61 \\ 32 & 42 & 51 & 57 & 68 & 80 & 79 & 67 \\ 48 & 61 & 63 & 65 & 74 & 66 & 68 & 65 \end{bmatrix}$$

(a) Source 1 and the quantization matrix

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 3 & 4 & 5 & 6 \\ 2 & 2 & 2 & 3 & 3 & 4 & 5 & 6 \\ 2 & 2 & 2 & 4 & 5 & 7 & 9 \\ 2 & 2 & 2 & 4 & 5 & 7 & 9 & 12 \\ 3 & 3 & 4 & 5 & 8 & 10 & 12 & 12 \\ 4 & 4 & 5 & 7 & 10 & 12 & 12 & 12 \\ 5 & 5 & 7 & 9 & 12 & 12 & 12 & 12 \\ 6 & 6 & 9 & 12 & 12 & 12 & 12 & 12 \end{bmatrix}$$

(b) Source 2 and the quantization matrix

$$\begin{bmatrix} 6 & 4 & 4 & 6 & 10 & 16 & 20 & 24 \\ 5 & 5 & 6 & 8 & 10 & 23 & 24 & 22 \\ 6 & 5 & 6 & 10 & 16 & 23 & 28 & 22 \\ 6 & 7 & 9 & 12 & 20 & 35 & 32 & 25 \\ 7 & 9 & 15 & 22 & 27 & 44 & 41 & 31 \\ 10 & 14 & 22 & 26 & 32 & 42 & 45 & 37 \\ 20 & 26 & 31 & 35 & 41 & 48 & 48 & 40 \\ 29 & 37 & 38 & 39 & 45 & 40 & 41 & 40 \end{bmatrix}$$

(c) Tampered image and the quantization matrix

Figure I-2. An example of image tampering involving JPEG double compression.

Regarding video data, due to the enormous amount of video data, they must be compressed before transmission and storage. Except some simple bit-stream splicing editing operations, most of video post-processing operations (such as filtering, adding scrolling texts, subtitles or other tampering operations) can deal with the content in the video scene only after these video streams have been decoded into image sequences, and finally these edited image sequences must be re-saved as compressed video files by the same or a different video encoder. Therefore, double compression is an indispensable link in the video post-processing, and the double compression detection technique is an effectual forensic tool to recover the processing history of digital video resources.

Some works have been presented to detect the forgery or related manipulation in audio streams, including MPEG-1 Audio Layer 3 or MP3 (Yang et al. 2008; Qiao, Sung and Liu 2010). For example, if two MP3 audio streams encoded at different bit-rates are selected in part and composited together and encoded in MP3 format, such forgery manipulation undergoes double MP3 compression. While we will be able to reveal the behavior of double MP3 compression, we may catch the forged part in MP3 audio streams. However, if two MP3 audio streams encoded at the same bit-rate and composited together and

encoded in MP3 format with the same bit-rate, the method of detecting double MP3 compression does not work.

Advanced Audio Coding (AAC), a lossy audio compression scheme, standardized by ISO and IEC, which was designed to be the successor of the MP3 format, generally obtains better sound quality than MP3 at similar bit rates. AAC is supported on iPhone, iPod, iPad, Nintendo DSi, iTunes, DivX Plus Web Player, PlayStation 3, PlayStateion Portable, Wii, Sony Walkman MP3, Sony Ericsson, Nokia, Android, Blackberry, and webOS-based mobile phones (Web AAC). While AAC audio files widely spread, to our knowledge, the literature of the forgery detection of AAC audio files is still missing to this date.

## Literature Citations and Review

*Steganography and forgery in JPEG images*

In steganography, quite a few steganographic algorithms have been proposed, including LSB embedding (Kurak and McHugh 1992), LSB matching (Mielikainen 2006), spread spectrum steganography (Marvel, Boncelet and Retter 1999), Outguess (Provos 2001), F5 (Westfeld 2001), model-based steganography (Sallee 2003 and 2005), Steghide (Hetzl and Mutzel 2005), BCH syndrome code based less detectable JPEG steganography (Sachnev, Kim and Zhang 2009), and highly undetectable steganography (HUGO) (Pevny, Filler and Bas 2010). Although these steganographic systems have been successfully steganalyzed (Chen and Shi 2008; Fridrich 2004; Fridrich, Kodovsky, Holub and Goljan 2011a, 2011b; Fu and Shi et al. 2006; Gu and Kurugollu 2011; Ker 2004; Kharrazi, Sencar and Memon 2006; Kodovsky and Fridrich 2009, 2011, 2012; Miche et al. 2009; Kodovsky, Pevny and Fridrich 2010; Kodovsky, Fridrich and Holub 2012; Li, Shi and Huang 2009; Liu, Sung and Ribeiro 2005; Liu et al. 2006; Liu Sung and Qiao 2008; Liu et al. 2008a, 2008b, 2008c; Liu and Sung 2007; Liu et al 2011a, 2011b, 2011c; Liu 2011a; Pevny and Fridrich 2007, 2008a and 2008b; Shi et al. 2007; Gul and Kurugollu 2011), the advances in steganography have posed new challenges to steganalyzers such as Gibbs construction-based steganography (Filler and Fridrich 2010), Syndrome-Trellis Codes based steganography (Filler, Judas and Fridrich 2011). Filler and Fridrich recently proposed a practical framework of adaptive steganographic systems which optimize the parameters of additive distortion functions and minimize the distortion for ±1 embedding in the DCT domain. This has greatly improved the art of hiding data in wide-spread JPEG images (Filler and Fridrich 2011). Yet Another Steganographic Scheme (YASS) was designed to be a secure JPEG steganographic algorithm with randomized embedding (Solanki, Sarkar and Manjunath 2007). By exploring the weakness of YASS steganographic system, Li, Shi and Huang (2009) presented a simple and efficient detection method by comparing the frequency of zero coefficients of the embedding host blocks and the neighboring blocks in DCT domain. This detection performance is very promising when the parameter of the big block (B-block) size is small (e.g., the size is set to 9 and 10). However, the detection performance apparently deteriorates if the parameter of B-block size increases (Li, Shi and Huang 2009). Kodovsky et al. designed 1234 features to detect YASS and tested 12 different configurations of YASS with a parameter of B-block size no larger than 11. In other words, the detection performance on the YASS

steganograms produced by a large parameter of B-block at 12, 13, 14, and 15 was missing (Kodovsky, Pevny and Fridrich 2010).

Regarding image forensics, the relevant manipulations, including double JPEG compression, source identification, image rescaling, copy-paste, inpainting, and compositing have been successfully detected (Pospescu and Farid 2004a, 2004b, 2005a, 2005b; Prasad and Ramakrishnan 2006; Alles et al. 2009; Bayram et al. 2005, 2006, 2008, 2009, 2010; Bianchi and Piva 2012a, 2012b; Celiktutan et al. 2008; Chang, Yu and Chang 2013; Chen and Hsu 2011; Chen, Fridrich et al. 2003, 2007, 2008; Fu, Shi and Su 2007; Choi et al 2006; Dirik et al. 2007; Gallagher 2005; Gou et al. 2007a, 2007b, and 2009; Gul and Avcibas 2009; Hsu and Chang 2009; Johnson and Farid 2005, 2006, 2007a, 2007b, 2007c; Lin et al. 2001 and 2005; Liu and Sung 2009; Liu, Sung and Qaio 2011c; Liu 2011b; Lukas and Fridrich 2003; Lukas et al. 2006; Mahdian and Saic 2008; Pan and Lyu 2010; Pan, Zhang and Lyu 2012; Shi et al 2007; Swaminathan et al. 2008). While most image forensics methods target traditional image tampering, seam carving-based image tampering in JPEG format has been ignored to some extent. Seam carving, an algorithm for image resizing, is known as content-aware scaling, liquid resizing or liquid rescaling and was designed by Shai Avidan of Mitsubishi Electric Research Labs (MERL) and Ariel Shamir of the Interdisciplinary Center and MERL. It establishes the paths of least importance in an image, called seams, automatically removes them and reduces the image size, or inserts seams to extend the image size (Avidan and Shamir 2007). Seam carving allows the removal of selected whole objects from photographs. The seam carving method for content-aware resizing and object removal has been implemented in Adobe Photoshop CS4 (Web photshop-cs4), GIMP (Web liquidrescale), digiKam (Web digikam), ImageMagick (Web imagemagick), as well as stand-alone programs such as iResizer (Web iresizer). The proliferation of seam-carved images presents a challenge to authorities who require image authentication. Sarkar et al.(2009) employed 324-dimensional Markov features, which was originally developed to detect JPEG-based steganograms by Shi et al. (2006), to distinguish between seam-carved, seam-inserted, and normal images. Fillion and Sharma designed a method which include benign image reduction, benign image enlargement, and deliberate image reduction to detect seam-carved images and tested their method over a set of images consisting of 1484 uncompressed images. Unfortunately, the JPEG images were not tested after content-aware manipulation (Fillion and Sharma 2010). The detection of seam-carving-based forgery in JPEG images needs extensive further studied.

*DCT-embedding-based adaptive steganography*

Most steganographic systems aim to minimize the distortion of the original cover. A practical framework to minimize statistical detectability when designing undetectable steganography was recently presented (Filler and Fridrich 2011). To design DCT-embedding-based adaptive steganography, an inter/intra-block cost model was given, as well as the performance of embedding algorithms based on the inter/intra-block cost model. The proposed DCT-embedding-based adaptive steganography was experimentally validated as being highly secure (Filler and Fridrich 2011). The embedding algorithms are optimized by using the multi-layered Syndrome-Trellis Codes (Filler, Judas and Fridrich

2011), with SVM and CC-PEV feature set (Kodovsky and Fridrich 2009), and Cross-Domain Feature set (Kodovsky and Fridrich 2011), respectively. The experiments show that proposed DCT-embedding-based adaptive steganography has greatly improved the state of DCT-embedding-based steganography (Filler and Fridrich 2011).

*YASS and A Detection Algorithm*

The original YASS algorithm presented in the reference (Solanki, Sarkar and Manjunath 2007). Although YASS embedding is not confined to the 8×8 block of the final JPEG compression, the location of embedding block in B-block is not random enough. By using QIM-based embedding, YASS also introduces additional zero DCT coefficients in the modified 8×8 block, and hence, the following algorithm was designed to break YASS (Li Shi and Huang 2009).

**Zero-value density-based approach to steganalysis of YASS (**Li, Shi and Huang 2009)

Transform a JPEG image under examination to spatial domain, denoted by $I_1$;

For $T = 9$ to 15

    For $s = 1$ to $T$

(a) Divide $I_s$ into non-overlapping consecutive $T{\times}T$ B-blocks;

(b) Collect 8×8 blocks from the upper left of all B-blocks and perform 2D DCT;

(c) Quantize the DCT coefficients by using $QF_a$;

(d) Compute the probability of zero rounded re-quantized DCT coefficients in candidate embedding bands and denote it by $Z_T(s)$;

(e) Crop the first $s$ columns and the first $s$ rows of $I_1$ to generate a new image $I_{s+1}$ for the next inner-loop;

  End

Compute the values of $\dfrac{1}{T-7}\sum_{i=1}^{T-7} Z_T(i)$ and $\dfrac{1}{7}\sum_{j=T-6}^{T} Z_T(j)$ as features.

End

As shown by this algorithm, the features are extracted from the candidate blocks along the diagonal direction of B-blocks, rather than from all possible 8×8 candidate blocks in B-blocks. In a B-block with the size of $T{\times}T$, there are a total of $(T-7){\times}(T-7)$ block candidates for embedding. Unfortunately, the above algorithm only selects the (T-7) blocks along a diagonal direction, not all candidate blocks, and as a result, the chance of the candidates along diagonal direction only hits 1/(T-7). While the value of T is large, the hit ratio is fairly low. For instance, if T=15, the hit ratio is only $1/8 = 0.125$. The experimental results

shown in the reference (Li, Shi and Huang 2009) also demonstrate that the detection accuracy is not satisfactory with a large T value.

In image forensics, the simplest way to get device information is extract that information from the header part of an image file. Most device vendors adopt EXIF (Exchangeable Image File Format) standard to write specific information into the header, like device maker, camera model, exposure, date and time the image captured, pixel size and etc. EXIF format is also handling sound and video record by digital camera, scanner, video machine, and other digital devices. If JPEG format is used to save images, quantization table can be obtained from JPEG images' header. Most manufacturers employ distinct quantization tables, some of them may define their own quantization tables. By examining the quantization table, we may simply indicate the origin device of the test image. However, this approach is less trustful in use on forensic evidence, because header information is very easy to be faked. Additionally, those properties may not available if image is resaved and recompressed.

Without EXIF information, another method to distinct given device is based on its image processing algorithm. Before image is saved to flash memory, the original data transferring from CCD (charge-coupled device) sensor need to be further processed, which include demosaicing, gamma correction, color processing, white point correction and last but not least compression. These post-processing is done with special DSP (digital signal processing) component of camera device. Although all manufacturers apply these general processing steps in their products, the processing detail and algorithm vary from one to another. Even with one vendor, the processing is different between distinct models. Therefore, it is supposed that output images contain some traits and pattern regardless of the original image content.

The source identification based on the different patterns of sensor noise/sensor finger print is successful, however once obtained images are processed again, for example, cropping, rescale (interpolation), and recompression, the identification generally becomes ineffective because the pattern of sensor noise is destroyed by the post-capture manipulation. Although several methods have been presented to detect single operation, e.g., cropping, image interpolation, double compression (Chen, Shi and Su 2008; Farid 1999, 2006, 2009; Liu and Sung 2009; Liu 2011b; Penvy and Fridrich 2008), it is still under-expressed to identify the camera source based on processed images with the combination of different operations.

With decreasing cost of mobile phone and megapixels camera phone quality increasing close to traditional digital camera, more and more people start using mobile phones to replace camera to capture pictures, since mobile phones are easily carried on and civilians can grasp scene easily and quickly. Many digital images shot by smartphones are widely spreading in society. However, smartphone-based image forensics is relatively ignored, compared to digital camera-based image forensics. Although Tsai et al. (2006, 2007) conducted such study to identify popular mobile phones with camera, in case the photographs were processed, such identification does not perform well. Moreover, if original images obtained from different semiconductor charge-coupled devices (CCD)

sensors are processed again, such as image resize, cropping or trans-coded to different image quality, these source identification algorithms become ineffective.

The images captured by smartphones are normally stored in JPEG format, not in raw format, to reduce the storage space. Nevertheless, the aforementioned steps are standard stages in digital images generated from camera pipeline and the exact processing detail in each stage varies from one producer from another. Different manufacturers may adopt their specific quantization tables to encode the captured digital signals into JPEG format. By recognizing these different quantization tables, it is usually easy for us to identify the source of smartphones, even that the exchangeable image file format (EXIF) data have been modified. However, if the EXIF data are removed and the images are manipulated again with different operations, the identification of the source and the revealing of the operations may become hard.

*MPEG double compression and detection*

When a MPEG-2 video is encoded in variable bit rate (VBR) mode that chooses a fixed quantization scale factor for the entire frame, the intra frame can be considered as a JPEG compressed image, and some features in detection of double JPEG compression are successfully introduced to detect double MPEG compression. Wang and Farid detected double MPEG-2 compression with variable bitrate (VBR) mode by examining the periodic artifacts introduced into the DCT histograms of I frames (Wang and Farid 2006) or modeling the marginal distribution of singly compressed and de-quantized DCT coefficients as a Gaussian distribution with expectation-maximization (EM) algorithm (Wang and Farid 2009). However, digital TV broadcasting, DVDs and Digital Video Recorders always adopt constant bitrate (CBR) mode to generate the MPEG-2 videos. Double MPEG-2 compression with CBR can be detected with Chen's approach which exploits the statistical disturbance in the first digit distribution of non-zero MPEG quantized AC coefficients (Chen and Shi 2008). Sun et al (2012) extended the features of Chen's algorithm to identify whether the bit rate of the secondary compression is bigger than that of the primary compression or not. But in their experiments, both the primary and the secondary MPEG-2 compression processes are implemented with the same MPEG-2 encoder. Their detection performance will decrease when a different MPEG-2 encoder is utilized to realize the secondary compression.

In the video compression system, the non-linear quantization is the mainly lossy coding technique to attain high compression ratios. Because some precision of DCT coefficients are lost in the quantization process, the de-quantization is not a fully reversible process of quantization. After a video sequence is compressed, there are some special traces in the distribution of reconstructed DCT coefficient. In the following section, Test Model 5 of MPEG-2 (ISO/IEC IS 13818-2; Test Model 5 for ISO/MPEG II) will be taken as an example to describe the quantization and de-quantization process in the MPEG-2 standard, and then two existing approaches are discussed to detect double MPEG-2 compression.

*Quantization and De-quantization process in MPEG-2*

Test Model 5 (TM5) is considered as a standard simulation platform for all researchers of MPEG-2, in order to verify the usefulness of various proposed coding techniques. In the module of quantization process of TM5, after each block is transformed with a 2-dimensional DCT coefficient matrix, the DCT coefficients are quantized respectively according to the mode of the current macro block.

In the intra mode, the quantizer of DC coefficients can only select a fixed step based on the precision parameter that will be transmitted in the picture coding extension. AC coefficients $ac(i, j)$ must be firstly quantized by individual quantization factors according to the following calculation.

$$qac(i, j) = \left\lfloor \frac{32 \times ac(i, j)}{w_I(i, j)} \right\rfloor; \qquad\qquad \text{(I-1)}$$

Where $\lfloor \cdot \rfloor$ is defined as integer division with truncation of the result toward zero. $w_I(i, j)$ is an intra-quantization weighting matrix, whose aim is to greatly reduce the amount of information in the high frequency components based on the characteristics of human visual system. An encoder can use the default weighting matrix in the MPEG-2 standard (shown as Figure I-3(a)); or introduce a new weighting matrix for some manufacturers (such as Figure I-3(b)), but this new matrix must be transmitted as a set of special parameters in the header of MPEG-2 stream file.

$$
\begin{bmatrix}
8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\
16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\
19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\
22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\
22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\
26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\
26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\
27 & 29 & 35 & 38 & 46 & 56 & 69 & 83
\end{bmatrix}
\qquad
\begin{bmatrix}
8 & 16 & 20 & 23 & 30 & 36 & 41 & 49 \\
16 & 20 & 23 & 28 & 36 & 41 & 48 & 52 \\
20 & 22 & 30 & 36 & 40 & 48 & 48 & 54 \\
22 & 26 & 34 & 37 & 41 & 48 & 52 & 66 \\
26 & 32 & 37 & 39 & 44 & 49 & 66 & 96 \\
34 & 35 & 41 & 42 & 49 & 66 & 102 & 106 \\
36 & 36 & 42 & 46 & 66 & 96 & 146 & 166 \\
38 & 40 & 48 & 64 & 96 & 106 & 166 & 206
\end{bmatrix}
$$
$$\text{(a)} \qquad\qquad\qquad\qquad\qquad \text{(b)}$$

Figure I-3. Intra quantization matrixes of two MPEG-2 encoders: (a) the default intra weighting matrix in TM5; (b) the intra weighting matrix in a DV.

The final quantized level $QAC(i,j)$ is calculated as follows:

$$QAC(i, j) = \left\lfloor \frac{qac(i, j) + sign(qac(i, j)) \times a \times q\_scale / b}{2 \times q\_scale} \right\rfloor. \qquad\qquad \text{(I-2)}$$

$$Sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0; \\ -1 & x < 0 \end{cases} \qquad\qquad \text{(I-3)}$$

The parameters $a=3$ and $b=4$ in TM5. $q\_scale$, i.e., quantization scale factor, is an important parameter to control the performance of quantization process.

In the non-intra mode, the DCT coefficients are quantized with a uniform quantizer that

has a dead-zone around zero. The quantization processes can be expressed as:

$$qac(i,j) = \left\lfloor \frac{32 \times ac(i,j)}{w_N(i,j)} \right\rfloor ; \tag{I-4}$$

$$QAC(i,j) = \left\lfloor \frac{qac(i,j)}{2 \times q\_scale} \right\rfloor . \tag{I-5}$$

For the de-quantization process in TM5, it strictly follows the MPEG-2 standard (ISO/IEC IS 13818-2 section 7.4). According to the weighting matrix $w(i,j)$ and $q\_scale$, de-quantization process can reconstruct DCT coefficients for all kinds of frames.

It is noticeable that the choose of $q\_scale$ is the principal method to implement different bit rate control schemes for meeting the requirement of different digital video applications. Since many video streaming are constrained by constant limited channel bandwidth (like 'DSL' or 'dial-up' connections) or fixed storage size ('Personal Video Player', or 'DVD Recorder'), CBR mode has been widely adopted because of its practical implementation, ease of use and flexibility over 'IP Networks'. To maintain the target output bit rate, $q\_scale$ will vary significantly among the different frames between 1 and 112 on a macroblock-to-macroblock basis, which is always directly calculated according to many factors (Wang 2000; Ding and Liu 1996), such as the status of the current buffer, bit allocation strategy, the spatial activity of the current macro block and so on. Compared with CBR mode, VBR mode can conserve the consistent visual quality due to a fixed $q\_scale$ for an entire frame, which has been extensively used in ATM-based broadband ISDN networks (Yu et al. 2001). As a result, the impact of $q\_scale$ on the distribution of reconstructed DCT coefficients is different between CBR and VBR mode, which indicates that double MPEG compression with CBR mode is distinct to double MPEG compression with VBR mode, as well to double JEPG compression.

Currently a wide variety of algorithms for detecting double JPEG compression have been reported in the literatures, but less attention has been paid to videos because of the complexity of video coding system. As the earliest algorithm, Wang and Farid (2006) exploited the static and temporal artifacts introduced by double MPEG-2 compression with VBR mode. In the spatial domain, as mentioned above, an intra-frame quantized by a constant quantization scale factor can be viewed as a JPEG image, and the features of double JPEG compression can be utilized to detect double MPEG-2 compression. For example, in Figure I-4, all frames in a test sequence (Figure I-4(a)) are doubly compressed by Berkeley MPEG Video Encoder (Mayer-Patel et al. 2005) with VBR and an obvious periodic artifact presents in the distribution of doubly quantized DCT coefficients (Figure I-4(b)). In the temporal domain, the frame deletion or insertion operations will induce large motion estimation errors at the following P frames and the periodic spikes in motion errors is utilized to detect frame tampering. However, the author has also discussed the limitation of this approach that it fails to detect the double MPEG compression with CBR mode, because CBR mode selects the quantization scale factor for each macro block and statistical features in doubly quantized DCT coefficients will be aliased by different quantization scale factors. When the same test sequence is single compressed by TM5

with CBR at 6 Mbps (Million bits per second), the histogram of quantization scale factors in intra frames is indicated in Figure I-4(c). It is found that the TM5 behaves as having a based quantization scale factor $q_B$ which is related to some initial conditions, such as output bit-rate, frame resolution, buffer size and so on. The others quantization scale factors are dynamically adjusted according to the spatial activity in the video scene. If the single compressed MPEG-2 video is doubly compressed at 7 Mbps, the distribution of doubly quantized DCT coefficients in intra frames is indicated as Figure I-4(d) where 'missing values', 'double peak' and 'periodic property' have disappeared, and it is necessary to find new features to detect double MPEG compression with CBR mode.



(a)
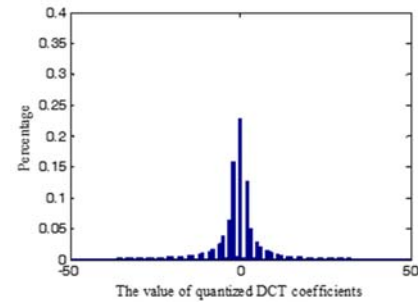


(b)
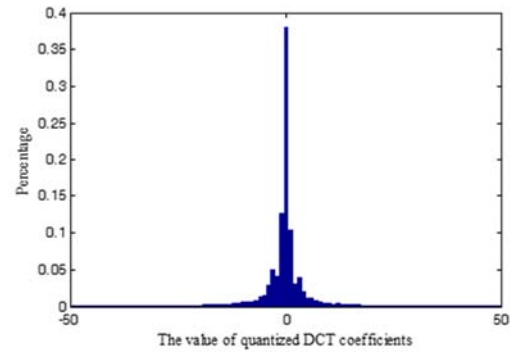


(c)

(d)

Figure I-4. Double MPEG-2 compression: (a) a standard test sequence from Video Quality Experts Group (VQEG); (b) the histogram of DCT (2, 1) coefficients in double MPEG-2 compressed intra frames with VBR mode; (c) the histogram of quantization scale factors in intra frames of the original MPEG compression with CBR mode; (d) the histogram of DCT (2, 1) coefficients in double MPEG-2 compressed intra frames with CBR mode.

Chen and Shi (2008) proposed another novel approach based on the first digit statistics (also called Bendford's Law) to detection of double MPEG-2 compression in both VBR and CBR videos. If the video is doubly compressed, the first digit distribution of non-zero MPEG-2 quantized AC coefficients in all kinds of frames will not meet the parametric logarithmic law. In order to make the detection more reliable, the GOP (group of picture) is proposed as the detection unit to obtain 36 features. In their experiment, the doubly

compressed video is also generated by the primary encoder, in other words, the primary and secondary coding processes adopt the same encoder. However, if the secondary encoder is different from the primary one, the first digit distribution of non-zero double quantized AC coefficients may obey to the generalized Bendford's Law again, and the detection performance of this algorithm will decrease.

*Audio steganalysis and forgery detection*

While most steganographic systems take digital images as carriers, digital audio files are also ideal as carrier for covert communication where the variety of audio encodings increases the difficulty of audio steganalysis. To detect the information-hiding in digital audio, Avcibas (2006) presented content-independent distortion measures as features for classifier design. Ozer et al. (2006) investigated the characteristics of the denoised residuals of audio files. Johnson et al. (2005) set up a statistical model by building a linear basis that captures certain statistical properties of audio signals. Zeng et al. (2008) presented a new algorithm to detect echo steganography based on statistical moments of peak frequency. Kraetzer and Dittmann (2008) proposed a Mel-cepstrum-based analysis to perform the detection of embedded hidden messages. Liu, Sung and Qiao (2009c) improved the performance of audio steganalysis by combining the Mel-cepstrum feature with a temporal derivative-based spectrum analysis. Geetha et al. (2010) presented high-order statistics of Hausdorff distance as discriminative features and investigated the application of evolving decision tree for audio steganalysis. Other authors' studies in audio steganalysis are included in references (Liu, Sung and Qiao 2009a, 2009c, and 2011b; Qiao, Sung and Liu 2009, 2010a and 2010b).

Although in the past years multiple steganalysis methods were designed to detect information-hiding in uncompressed audio, the information-hiding in compressed audio, such as MPEG-1 Audio Layer 3, more commonly referred as MP3, has been barely explored due to the complexity and variety of the compression algorithms. As a result of the different characteristics between compressed and uncompressed audio, most existing methods do not work for steganalysis of audio in the compression domain, and the decompression attempt, which erases the hidden data through the de-quantization step in signal reconstruction, leads to a failure of those methods on decompressed audio.

As one of the most popular audio formats on the Internet, MP3 provides a faithful reproduction of the original signal with a small amount of data. The widespread use and flexible encoding algorithm enable it to be a desirable carrier for covert communication. Böhme and Westfeld (2004) investigated the characteristics of MP3 encoders for potential applications in steganography or steganalysis. Although different encoders are designed to be compatible with the MP3 standard, statistical analysis also illustrates the distinctions among available MP3 encoders. MP3Stego (Web MP3Stego) is one of the most widely used audio steganographic tools, especially for MP3 audio. MP3Stego is implemented by combining a novel information-hiding algorithm with an existing MP3 encoder. MP3Stego (Web MP3Stego) is built on the MP3 encoder and decoder from 8 Hz and ISO MPEG Audio Subgroup Software Simulation Group, respectively. All payloads are encrypted using 3DES and then embedded in frames randomly selected by using SHA-1.

With uncompressed waveform audio (WAV) as input, MP3Stego embeds data during the encoding process and generates a steganogram in MP3 format. The algorithm of MP3Stego exploits the audio degradation from lossy compression and embeds data by slightly expanding the distortion of the signal without attracting attention from the listener. MP3Stego embeds compressed and encrypted data in an MP3 bit stream during the compression process. In the heart of layer 3 compression, two nested loops manipulate the trade-off between file size and audio quality. The hiding process occurs in the inner loop where the quantization step size is increased to fit the available number of bits.

Regarding audio forgery detection, while Yang et al. (2008) designed a method to check the offset in MP3 frame compression artifact to detect MP3 forgery, as designed a successor to MP3 compression method, AAC audio files have been widely disseminated. However, to our knowledge, the literature of the forgery detection of AAC audio files was still missing before we explored it in this project.

**Statement of Hypothesis or Rationale for the Research**

*Hypothesis 1. Most information hiding and forgery manipulations in JPEG images modify the statistics of DCT coefficients including the neighboring joint density and the calibrated versions*

*Hypothesis 2. JPEG double compression modifies the marginal density and neighboring joint density in DCT domain*

*Hypothesis 3. In MPEG double compression, the distribution of reconstructed DCT coefficients after double compression will be different from that of original MPEG video*

*Hypothesis 4. MP3stego will modify the statistics of frequency-based subband moment statistics, accumulative neighboring joint probabilities and accumulative Markov transition probabilities in the compression domain*

*Hypothesis 5. AAC audio forgery will change the original frame compression structure and hence leave a clue for the detection*

# II. Methods

## II-1. JPEG Steganalysis

### *II-1-a. Algorithm Design*

Inspired by a multivariate generalized Gaussian distribution (MGGD) model in the wavelet that was successfully used for image denoising (Cho and Bui 2005), we discussed the MGGD in the DCT domain and pointed out that approximate distribution of neighboring joint density of DCT coefficients may be modeled by MGGD, and information-hiding generally affects the distribution (Liu Sung and Qiao 2009a, 2009b; Liu Sung and Qiao 2011a). Our study also shows that besides information hiding, JPEG-based double compression and interpolation modify the neighboring joint density also and hence leave a clue to reveal the manipulations (Liu, Sung and Qiao 2008; Liu, Sung, Riberio and Ferreira 2008; Liu, Sung and Qiao 2009a, 2011a, 2011b; Qiao, Sung and Liu 2013). Our experimental results indicate that neighboring joint density-based approach outperforms the Markov transition probability-based approach in JPEG steganalysis. We analyzed the reason that neighboring joint density-based approach is generally superior to highly referenced Markov-based approach: "it is the modification of the neighboring joint density that results in the modification of Markov transition probability" (Liu, Sung and Qiao 2011a). We can completely derive Markov transition probability from neighboring joint density, but we cannot derive the neighboring joint density from Markov transition probability, in other words, neighboring joint density contains more discriminant information compared to Markov transition probability.

Normally, neighboring joint density of DCT coefficients is symmetric to the origin. Our previous detection algorithm (Liu, Sung and Qiao 2011a) is designed in the following:

### *Neighboring Joint Density on Intra-block*

Let F denote the quantized DCT coefficient array consisting of M×N blocks $F_{ij}$ ($i = 1, 2, …, M$; $j = 1, 2, …, N$). The intra-block neighboring joint density matrix on horizontal direction $absNJ_{1h}$ and the matrix on vertical direction $absNJ_{1v}$ are given by:

$$absNJ_{1h}(x,y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{8}\sum_{n=1}^{7}\delta(|c_{ijmn}| = x, |c_{ijm(n+1)}| = y)}{56MN} \tag{II-1}$$

$$absNJ_{1v}(x,y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{7}\sum_{n=1}^{8}\delta(|c_{ijmn}| = x, |c_{ij(m+1)n}| = y)}{56MN} \tag{II-2}$$

Where $c_{ijmn}$ is the DCT coefficient located at the $m^{th}$ row and the $n^{th}$ column in the block $F_{ij}$; $\delta = 1$ if its arguments are satisfied, otherwise $\delta = 0$; $x$ and $y$ are integers. For

computational efficiency, we define *absNJ₁* as the neighboring joint density features on intra-block, calculated as follows:

$$absNJ_1(x, y) = \{absNJ_{1h}(x, y) + absNJ_{1v}(x, y)\} / 2 \qquad \text{(II-3)}$$

In our prior detection, the values of $x$ and $y$ are in the range [0, 5], and *absNJ₁* consists of 36 features.

*Neighboring Joint Density on Inter-block*

The inter-block neighboring joint density matrix on horizontal direction absNJ$_{2h}$ and the matrix on vertical direction absNJ$_{2v}$ are constructed as follows:

$$absNJ_{2h}(x, y) = \frac{\sum_{m=1}^{8}\sum_{n=1}^{8}\sum_{i=1}^{M}\sum_{j=1}^{N-1}\delta(|c_{ijmn}| = x, |c_{i(j+1)mn}| = y)}{64M(N-1)} \qquad \text{(II-4)}$$

$$absNJ_{2v}(x, y) = \frac{\sum_{m=1}^{8}\sum_{n=1}^{8}\sum_{i=1}^{M-1}\sum_{j=1}^{N}\delta(|c_{ijmn}| = x, |c_{(i+1)jmn}| = y)}{64(M-1)N} \qquad \text{(II-5)}$$

We define absNJ$_2$ as the neighboring joint density features on inter-block, calculated as follows:

$$absNJ_2(x, y) = \{absNJ_{2h}(x, y) + absNJ_{2v}(x, y)\} / 2 \qquad \text{(II-6)}$$

Similarly, the values of x and y are in [0, 5] and absNJ$_2$ has 36 features. In our previous approach, the neighboring joint density features defined by equations (II-3) and (II-6) are denoted by *absNJ*, containing 72 features (Liu, Sung and Qiao 2011a).

CALIBRATED NEIGHBORING JOINT DENSITY

We have shown and validated the modification of the neighboring joint density caused by information hiding of several DCT-embedding steganographic systems (Liu, Sung and Qiao 2011a). Regarding DCT-embedding adaptive steganography that aims to minimize the distortion cost through Syndrome-Trellis Codes (Filler and Fridrich 2010), although the modification is very small, it does change the neighboring joint density (Figure II-1). Figure II-1(a) shows a JPEG cover. Figure II-1(b) gives the JPEG steganogram produced by using DCT-embedding-based adaptive hiding algorithm (Filler and Fridrich 2010) with the relative payload of 0.4 bits per non-zero-AC (bpac). Figure II-1(c) demonstrates the difference of the intra-block-based neighboring joint density when comparing the cover and the steganogram and Figure II-1(d) shows the difference of the neighboring joint density of the absolute array of DCT coefficients when comparing the cover and the steganogram.

(a)      JPEG cover               (b)      JPEG steganogram

(c)      Difference of NJ density         (d)      Difference of absNJ density
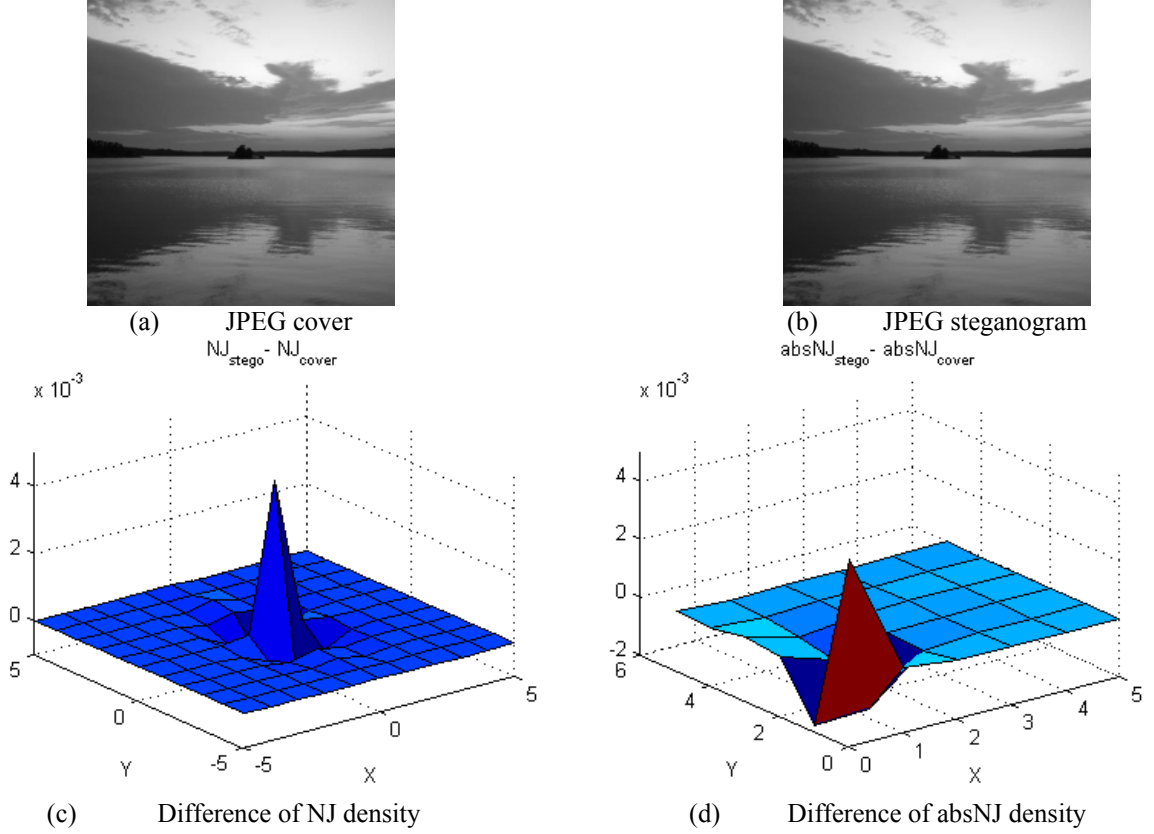
Figure II-1. An example to demonstrate the modification of neighboring joint (NJ) density features by DCT-embedding-based adaptive steganography.

It should be noted that we do not have the original cover as a reference while detecting steganography. For example, in Figure II-1, only given the JPEG image (a) or (b), not both, we need to determine whether the image under examination is a cover or a steganogram; it is impossible for us to obtain the density difference shown in (c) and (d) in real detection. We should also mention that the neighboring joint density varies across different JPEG images. Therefore, there are still limitations to detecting the steganogram if we only adopt the neighboring joint density feature set without any reference, originally presented in the references (Liu, Sung and Qiao 2009a and 2011a).

To capture the modification of the density caused by data embedding, suggested by the self-calibration that was presented in (Fridrich 2005) and based on our previous steganalysis method (Liu, Sung and Qiao 2009a and 2011a), we design a calibrated neighboring joint density-based approach, described as follows:

*a.* The neighboring joint density features $absNJ_1(x, y)$ and $absNJ_2(x, y)$, defined by equations (5) and (8), are extracted from a JPEG image under examination;

*b.* The testing JPEG image is decoded to spatial pixel values and cropped by $i$ rows and $j$ columns ($0 \leq i < 7$, $0 \leq j < 7$, and $i+j > 0$). The cropped image is encoded in JPEG format with the same quantization matrix, and the joint density features denoted by

$absNJ_{1i,j}^{c}(x,y)$ and $absNJ_{2i,j}^{c}(x,y)$ are extracted from the cropped and recompressed JPEG images, here $(i,j) \in \{(0,1),(0,2),...,(1,0),(1,1),...,(7,7)\}$;

***c.*** The mean values of $absNJ_1^c(x,y)$ and $absNJ_2^c(x,y)$ are calculated by

$$\overline{absNJ_1^c}(x,y) = \frac{1}{63} \sum_{(i,j)} absNJ_{1i,j}^c(x,y) \tag{II-7}$$

$$\overline{absNJ_2^c}(x,y) = \frac{1}{63} \sum_{(i,j)} absNJ_{2i,j}^c(x,y) \tag{II-8}$$

***d.*** The differential joint density features are given by

$$absNJ_1^D(x,y) = \overline{absNJ_1^c}(x,y) - absNJ_1(x,y) \tag{II-9}$$

$$absNJ_2^D(x,y) = \overline{absNJ_2^c}(x,y) - absNJ_2(x,y) \tag{II-10}$$

In our detection, we either adopt the neighboring joint density features, given by equations (II-3) and (II-6), and the reference density features, given by equations (II-7) and (II-8), together as a detector, or adopt the features defined in equations (II-3), (II-6), (II-9), and (II-10) together as a detector. We should note that both detectors are actually the same, because each one feature set can completely be derived from another. Our experiments also verify that both feature sets have approximately identical detection performance by using different classifiers. By using a fisher linear discriminant and logistic regression classifier, especially, we have obtained exactly same detection results. The detector of calibrated neighboring joint density containing the features is denoted by CC-absNJ.

To demonstrate the effectiveness of a calibrated neighboring joint density-based approach, Figure II-2 (a) shows a JPEG cover image, Figure II-2(b) plots the neighboring joint density defined in (II-3), and Figure II-2(c) manifests the differential joint density, defined in equation (II-9). Figure II-2 (d) shows the JPEG steganogram produced by F5 algorithm, Figure II-2(e) is the neighboring joint density defined in (II-3), and Figure II-2(f) gives the differential joint density defined in equation (II-9). The original neighboring joint density from cover and the density from steganogram are different, as are the differential joint densities.

Figure II-3 (a) shows a JPEG cover image and Figure II-3(d) presents the steganogram produced by using adaptive-embedding algorithm (Filler and Fridrich 2010). Original neighboring joint densities from the cover and from the steganogram are given in (b) and (e) respectively, and the differential densities are plotted in (c), and (f), respectively. The difference of the self-differential density between the cover and the steganogram is noticeable.

Figure II-2(a) and Figure II-3(a) also demonstrate that different JPEG images have different neighboring joint densities (Figure II-2(b) and Figure II-3(b)), implying the importance of self-differential density for steganalysis.
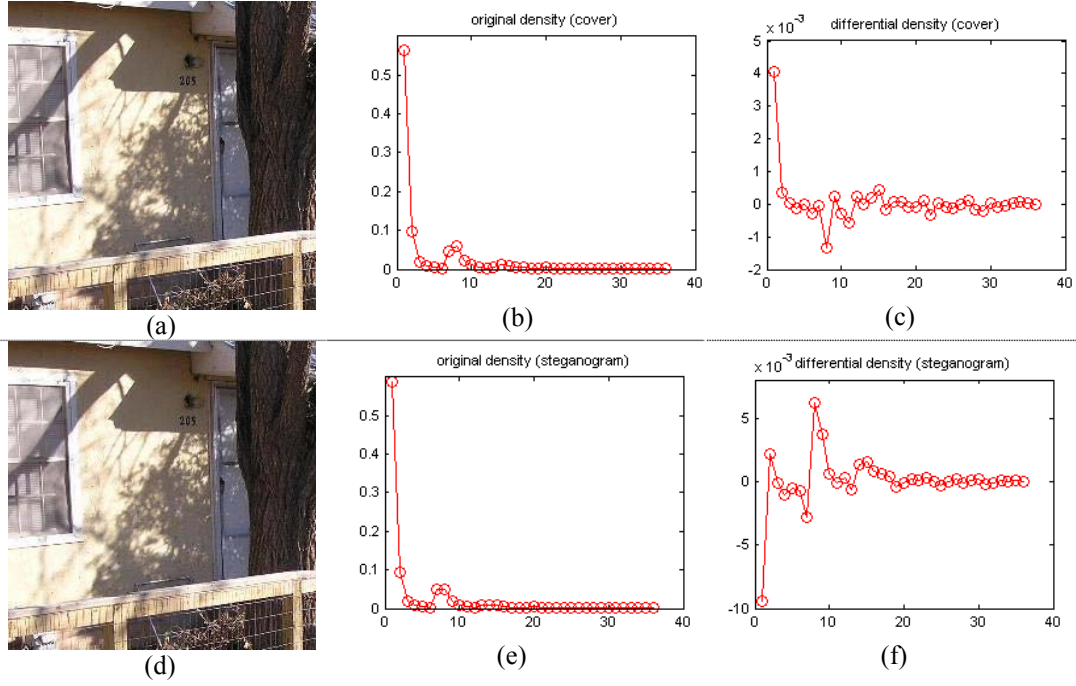
Figure II-2. A demonstration of a JPEG cover image (a) and the F5 steganogram (d). Original neighboring joint densities are shown in (b) and (e), and the self-differential densities are given in (c), and (f), respectively.
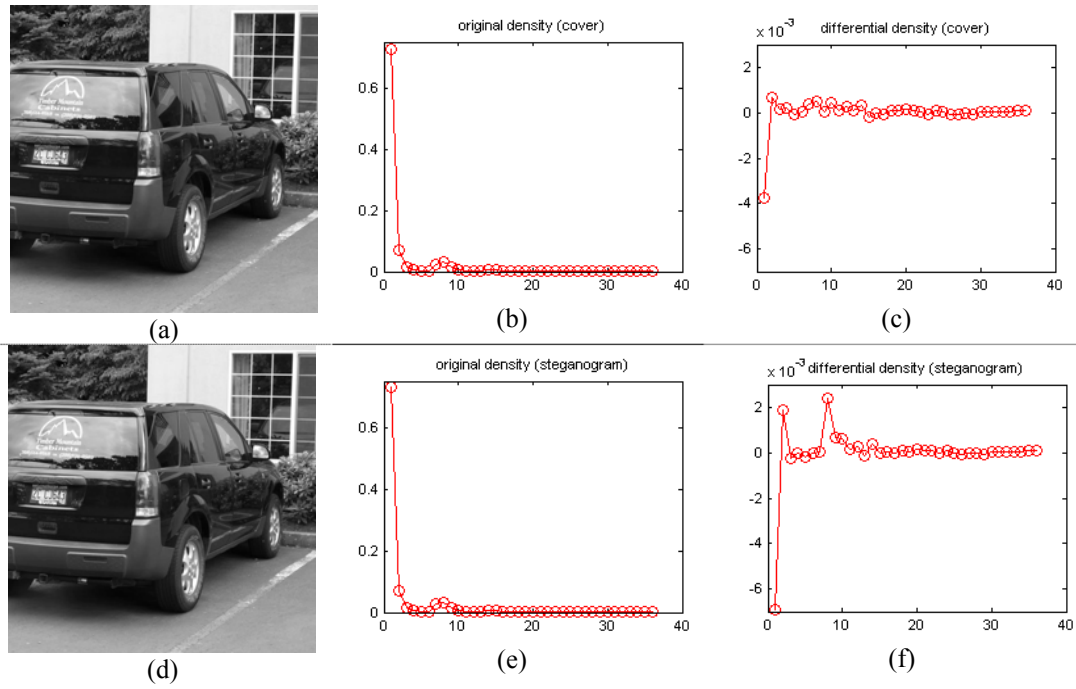


Figure II-3. A JPEG cover image (a) and the adaptive-embedding steganogram (d). Original neighboring joint densities are shown in (b) and (e), and the self-differential densities are given in (c), and (f), respectively.

*II-1-b. Experiment Design*

*Materials*

The 5000 original color TIFF raw format digital images used in the experiments are 24-bit, 640×480 pixels, lossless true color, never compressed. We cropped these original images into 256×256 pixels in order to eliminate the low complexity parts and converted the cropped images into JPEG format with the default quality, the same to pervious steganalysis experimental setup (Liu, Sung et al. 2006; Liu and Sung 2007; Liu, Sung, Chen and Xu 2008; Liu, Sung and Qiao 2009b; Liu, Sung et al. 2010). The following steganograms are generated with different hiding ratios, measured by relative payload, or the ratio of the number of DCT-coefficients modified to the total number of non-zero valued AC DCT-coefficients.

**F5** – Westfeld (2001) proposed the algorithm F5 that withstands visual and statistical attacks, yet it still offers a large steganographic capacity.

**Steghide** – Hetzl and Mutzel (2005) designed a graph-theoretic approach for information-hiding based on the idea of exchanging rather than overwriting pixels. Their approach preserves first-order statistics, and the detection on the first order does not work.

**Model Based steganography without deblocking (MB1)** – Sallee (2003) presented an information-theoretic method for performing steganography. Using the model-based methodology, an example steganography method is proposed for JPEG images which achieves a higher embedding efficiency and message capacity than previous methods, while remaining secure against first order statistical attacks.

**Model Based steganography with deblocking (MB2)** – Based on model-based steganography, Sallee (2005) presented a method to defend against "blockiness" steganalysis attack.

**Adaptive steganography in JPEG images** – In order to produce DCT-embedding-based adaptive steganography, 1000 BOSSRank cover images downloaded from (Web Boss) are converted into JPEG images first at the quality factor "75". The adaptive steganograms are produced by using the adaptive DCT-embedding hiding tool (Filler and Fridrich 2010), and the parameter of hiding bits per non-zero-AC (bpac) is set from 0.1 to 0.35 with the step size of 0.05 bpac.

*Detectors and learning classifiers*

In our study, the following steganalysis detectors are compared, including: 1) 72-dimensional *absNJ*, neighboring joint density-based JPEG steganalysis originally designed in the reference (Liu Sung and Qiao 2011a); 2) 144-dimensional **CC-absNJ,** calibrated neighboring joint density, consisting of 144 features, defined by (II-3), (II-6), (II-7), and (II-8), or by (II-3), (II-6), (II-9), and (II-10). We argue that both 144-dimensional feature sets are actually identical in terms of the detection capability; 3) 548-

dimensional ***CC-PEV*** (Kodovsky and Fridrich 2009); 4) 274-dimensional ***PEV*** (Penvy and Fridrich 2007); 5) 486-dimensional **Markov**-process-based detector (Chen and Shi 2008); 6) 48,600-dimensional rich model ***CC-C300***, a high-dimensional rich model for JPEG steganalysis (Kodovsky and Fridrich 2011); 7) 7,850-dimensional compact rich model ***CF*** for JPEG steganalysis (Kodovsky, Fridrich and Holub 2012); 8) 22,510-dimensional Cartesian calibrated JPEG domain rich model ***CC-JRM*** (Kodovsky and Fridrich 2012); and 9) a union of spatial domain rich model with the fixed quantization q=1c, 12,753-dimensional *SRMQ1* (Fridrich and Kodovsky 2012), and 22,510-dimensioal *CC-JRM*, denoted by ***CC-JRM+SRMQ1***, a total of 35,263 features (Kodovsky and Fridrich 2012). Table II-1 lists these detectors and the feature dimensionality.

Table II-1. Steganalysis detectors examined in our study

| Detector | Feature dimensionality | Reference |
|---|---|---|
| CC-absNJ | 144 | {(II-3),(II-6),(II-7),(II-8)} or {(II-3), (II-6), (II-9), (II-10)} |
| absNJ | 72 | (Liu Sung and Qiao 2011a) |
| CC-PEV | 548 | (Kodovsky and Fridrich 2009) |
| PEV | 274 | (Penvy and Fridrich 2007) |
| Markov | 486 | (Chen and Shi 2008) |
| CC-C300 | 48600 | (Kodovsky and Fridrich 2011) |
| CF | 7850 | (Kodovsky, Fridrich and Holub 2012) |
| CC-JRM | 22150 | (Kodovsky and Fridrich 2012) |
| CC-JRM+SRMQ1 | 35263 | (Kodovsky and Fridrich 2012; Fridrich and Kodovsky 2012) |

Support Vector Machines (SVM) (Chang and Lin 2011; Vapnik 1998), Fisher's Linear Discriminant (FLD) to minimize the errors in the least square sense (Heijden et al 2004), and an ensemble classifier that was used with rich models for steganalysis (Kodovsky, Fridrich and Holub 2012) are employed in our comparison study. It should be noted that the computational cost by SVM is too high for rich models due to the high dimensionality of the feature set, and rich model-based steganalysis detectors are not suitable with SVM. However, the low-dimensional detectors proposed in our study are easily utilized with SVM.

To select SVM for the low-dimensional detectors, we compare the popular algorithms LibSVM (Chang and Lin 2011), SVM_light (Joachims 2002), the SVM algorithms implemented in PRtools (Heijden et al 2004), and five SVM learning algorithms in LIBLINEAR (Fan et al 2005). We compare these SVM algorithms with different parameters including linear, polynomial, and radial basis function (RBF) kernels. In our comparison, although the algorithms implemented in LIBLINEAR package are the fastest, the accuracy is the lowest LibSVM generally obtains the best detection accuracy. Therefore,we finally employed LibSVM with optimal kernel parameters after comparing different combinations of kernel parameters by grid search (Chang and Lin 2011).

While we apply the ensemble classifier that was used in (Kodovsky, Fridrich and Holub 2012), the optimized parameters are computed first, including the optimization of the sub-

dimensionality and optimal base learning classifiers. By optimizing the parameters and applying optimized ensemble classifier to rich model-based detectors, the computational cost, is much higher than if using Fisher linear discriminant.

We perform one hundred experiments for each feature set at each hiding ratio by using each classifier. In each experiment, 70% of the samples are randomly selected for training, and the other 30% samples are used for testing. The prediction outcomes on testing data can be divided into True Negative (TN), False Negative (FN), False Positive (FP), and True Positive (TP). Without losing a generality, our detection accuracy is calculated by 0.5*TN/(TN+FP)+0.5*TP/(TP+FN).

## II-2. YASS STEGANALYSIS

### II-2-a. Algorithm design

By searching all possible 8×8 candidate blocks in B-blocks, we extract the neighboring joint density of the DCT coefficients from all candidate blocks that are possibly used to carry hidden data and the 8×8 non-candidate block neighbors that are not sued for information hiding and then calculate the difference of the joint density values of the candidates and the non-candidate neighbors. Our algorithm of feature design to detect YASS steganogram is described as follows:

Starting from the large B-block parameter T = 9,

1. Decode the JPEG image under scrutiny to spatial domain and divide it into non-overlapping consecutive $T \times T$ B-blocks;

2. In each $T \times T$ B-block, search all 8×8 blocks possibly used for information hiding, in a total of $(T-7)^2$ candidate blocks. The set of all candidate blocks of the image under detection is denoted by CB. For each candidate block CB($i$) ($i$=1,2, …, $CN$, $CN$ is the number of all candidate blocks on the testing image); subtract 128 from each pixel value, then apply two-dimensional DCT transform, quantize the DCT coefficients by using the quantization matrix corresponding to $QF_a$, and obtain the absolute DCT coefficient array. The neighboring joint density features on the intra-block of CB($i$), denoted by $absNJ(i; x,y)$, is given by

$$absNJ(i;x,y) = 0.5 \times \left( \frac{\sum_{m=1}^{8}\sum_{n=1}^{7} \delta\left(\left|c_{mn}^i\right| = x, \left|c_{m(n+1)}^i\right| = y\right)}{56} + \frac{\sum_{m=1}^{7}\sum_{n=1}^{8} \delta\left(\left|c_{mn}^i\right| = x, \left|c_{(m+1)n}^i\right| = y\right)}{56} \right) \quad \text{(II-11)}$$

Where $c_{mn}^i$ is the DCT coefficient located at the $m^{\text{th}}$ row and the $n^{\text{th}}$ column in the candidate block CB($i$); $\delta = 1$ if its arguments are satisfied; otherwise $\delta = 0$; $x$ and $y$ are integers.

3. From all 8×8 blocks that are adjacent to the candidate block CB(*i*) in the horizontal/vertical direction but without any overlap to CB(*i*), the adjacent 8×8 blocks that do not belong to CB are denoted by NC(*i,j*). Generally, non-candidate 8×8 blocks must be across two adjacent T×T B-blocks, such as when a *T×T* B-block is not on the boundary or on the corner of an image under examination,

(a) if an 8×8 block candidate is located inside of the B-block without any overlap to the B-block boundary, it has four non-candidate neighbors, shown by Figure II-4(a);

(b) if an 8×8 block candidate overlaps at only one of the four boundary borders of the B-block, it has three non-candidate neighbors, shown by Figure II-4(b);

(c) if an 8×8 block candidate overlaps at two of the four boundary borders of the B-block or is located at one of four corners of the B-block, it has two non-candidate neighbors, shown by Figure II-4(c).



(a)                          (b)                          (c)
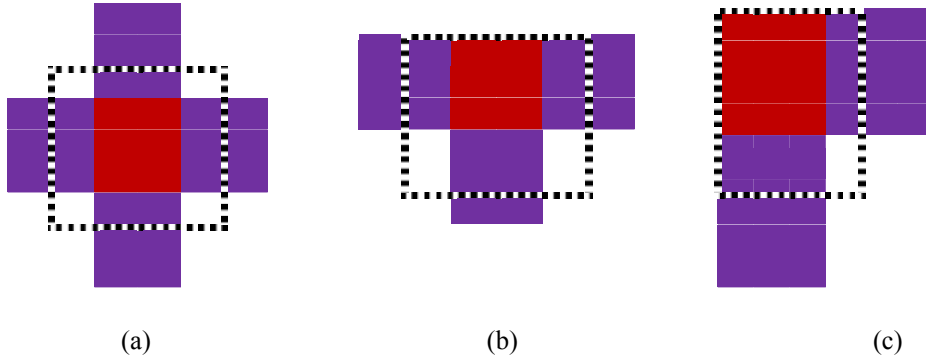
Figure II-4.  A candidate block is located in a B-block (dashed), and the non-candidate neighbors are across two B-blocks.

4. The neighboring joint density on the non-candidate neighboring block NC(i,j) is given by

$$absNJ(i, j; x, y) = 0.5 \times \left( \frac{\sum_{m=1}^{8}\sum_{n=1}^{7} \delta\left(\left|c_{mn}^{ij}\right| = x, \left|c_{m(n+1)}^{ij}\right| = y\right)}{56} + \frac{\sum_{m=1}^{7}\sum_{n=1}^{8} \delta\left(\left|c_{mn}^{ij}\right| = x, \left|c_{(m+1)n}^{ij}\right| = y\right)}{56} \right)$$  (II-12)

Where $c_{mn}^{ij}$ is the DCT coefficient located at the $m^{th}$ row and the $n^{th}$ column in the non-candidate block NC(*i,j*). $\delta = 1$ if its arguments are satisfied; otherwise $\delta = 0$; *x* and *y* are integers.

5. The mean value of the differential neighboring joint density between candidate blocks and non-candidate blocks are given by

$$diff - absNJ(x, y) = \frac{\sum_i absNJ(i; x, y)}{count(CB)} - \frac{\sum_{(i,j)} absNJ(i, j; x, y)}{count(NC)}$$ (II-13)

Where count (CB) gives the total number of candidate blocks, and count (NC) gives the total number of non-candidate blocks on the testing image.

The features defined in equation (II-13) constitute the feature set to detect the YASS steganogram produced by large B-block size T. The values of x and y are set from (0,0), (0,1), (0, 2), (1, 0), ...to (2,2), in a total of 9 differential neighboring joint density features for a single value of B-block size T.

6.     While T < 16, set T+1 to T, repeat 1 to 6.

The final detector contains 63 differential features for all possible T parameters (T = 9, 10, ... 15).

Figure II-5 shows a cover and YASS steganograms produced with B-block size of 9, 11, and 13 on the left. The diff-absNJ features extracted from the cover and the steganograms are shown on the right, manifesting different patterns between the cover and different steganograms produced by different B-block size.



cover



YASS steganogram (T=9)
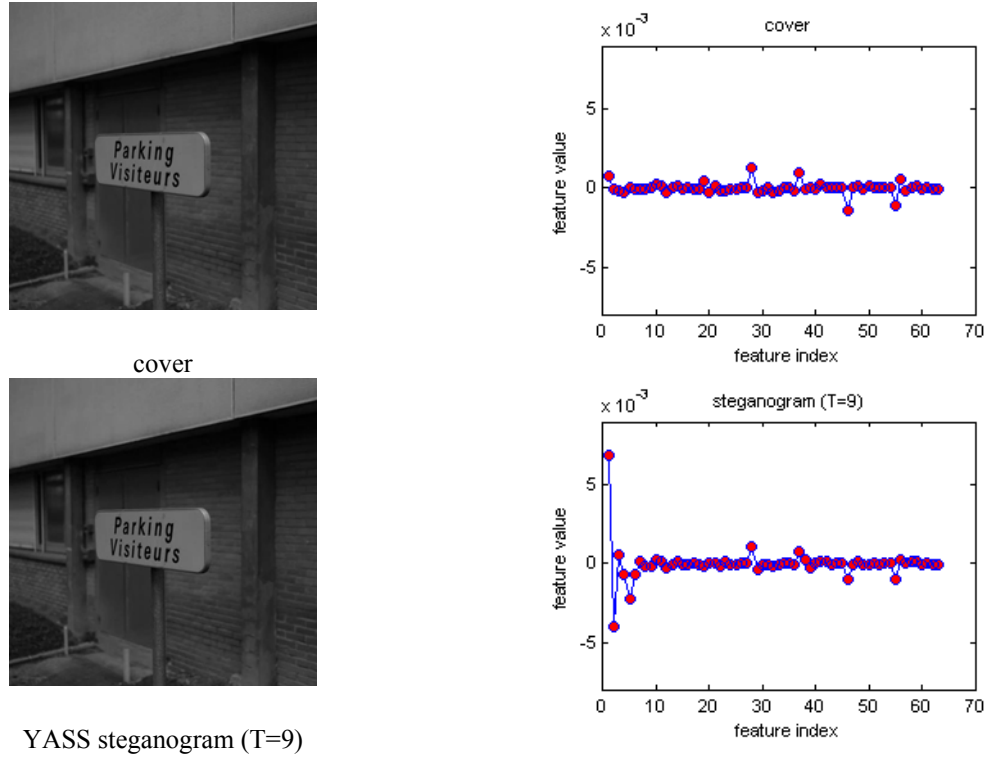
YASS steganogram (T=11)



YASS steganogram (T=13)
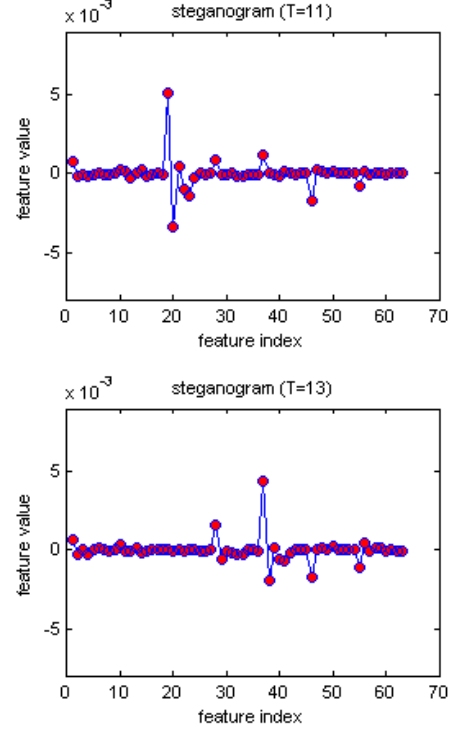
Figure II-5. Different patterns of diff-absNJ features among cover image and YASS steganograms ($QF_h = QF_a = 75$) with B-block parameter T=9, 11, and 13. The cover and steganograms are shown on the left and the diff-absNJ features on the right.

## II-2-b. Experimental design

The original 1000 BOSSRank cover images downloaded from (Web Boss) are used for YASS embedding. We set $QF_h = QF_a = 75$ in production of the steganograms. Accordingly, we encode the 1000 BOSSRank cover images in JPEG format at the quality factor of 75 as cover images. To create YASS steganograms, $QF_h$ and $QF_a$ are set to the same quantization factor in order to avoid double JPEG compression in YASS steganograms. If $QF_h$ is not equal to $QF_a$, the YASS steganograms could be detected by exposing the double JPEG compression. Double JPEG compression has been documented with very good detection performance (Chen and Hsu 2011; Liu Sung and Qiao 2011c; Liu et al 2013). Additionally, the big B-block size T is set from 9 to 15 respectively to produce the steganograms.

To conduct a comparative study, we extract the *diff-absNJ* features (Liu 2011a)), and the zero-value density features in the reference (Li, Shi and Huang 2009). LibSVM and FLD classifiers are used for classification. In each experiment, 50% samples are randomly selected for training, and the other 50% samples are used for testing; 200 experiments are operated for each feature set at each B-block size by using each learning classifier for binary classification, and 200 experiments are conducted for each feature set by mixing covers and all YASS steganograms together for multiple-class classification. Our

approach and zero-value density -based detection are based on the exposure of potential candidate blocks for data hiding. Unlike zero-value density-based approach, our method does not assume the embedding position on the first few positions in the candidate blocks. By using an ensemble classifier and FLD, we also employ the union of CC-JRM and SRMQ1 (Kodovsky and Fridrich 2012; Fridrich and Kodovsky 2012), a 35263-deminsional feature set to detect steganograms without exposing the candidate blocks that are used for embedding.

## II-3. SEAM-CARVED FORGERY DETECTION IN JPEG IMAGES

### II-3-a. Algorithm design by integrating calibrated neighboring joint density with spatial rich models

In seam carving, finding the seam is completed with the path of minimum cost from one end of the image to another. While seam carving allows for removal of selected whole objects from photographs or removing/inserting some seams, the manipulation occurs in spatial domain–it directly modifies the pixel values in the spatial domain. In addition to altering the pixel values, the removal or insertion of seams also results in the change of some pixel positions in the original image and in destroying the original compression block structure, hence leaving the trace of the manipulation both in the spatial domain and in the transform domain. Based on these facts, we use calibrated neighboring joint density features that have been described previously to reveal the modification in the transform domain. To keep track of the modification in the spatial domain, we directly make use of a spatial domain rich model, recently designed for steganalysis (Fridrich and Kodovsky 2012), to capture the modification of the statistical features. We surmise that the spatial domain rich model may be very effective in detecting the seam-carving-based manipulation in the spatial domain since seam carving directly removes/inserts seams in the spatial domain and changes the pixel values and positions. In addition to the comparison of the detection performance of the calibrated neighboring joint density in the DCT domain and spatial rich model based features in the spatial domain, we integrate these two types of feature sets together for the detection with the expectation of obtaining better detection accuracy.

Figure II-6 shows an example to verify the modification of the joint density in DCT domain and the modification of the pixel values in grayscale format on the red, green, and blue channels. An untouched JPEG image and the forged JPEG image by seam carving are shown in (a) and (d), respectively. In the forgery, the image of the man at the center of the original photo has been removed. The neighboring joint densities in the DCT domain directly extracted from the untouched image and from the tampered image are given in (b) and (e), and the differential densities between original density and the calibrated density are given in (c), and (f), respectively. To reveal the modification in the spatial domain, Figure II-6(g) gives the difference of the grayscale values between the tampering and the untouched photo, Figure II-6(h). Figure II-6(i) and Figure II-6 (j) demonstrate the difference of the pixel values on red, green and blue channels,

respectively; the tampering has noticeably modified the pixel values. Some modifications go as high as 200, implying that the spatial domain-based feature set could be very effective for the detection.
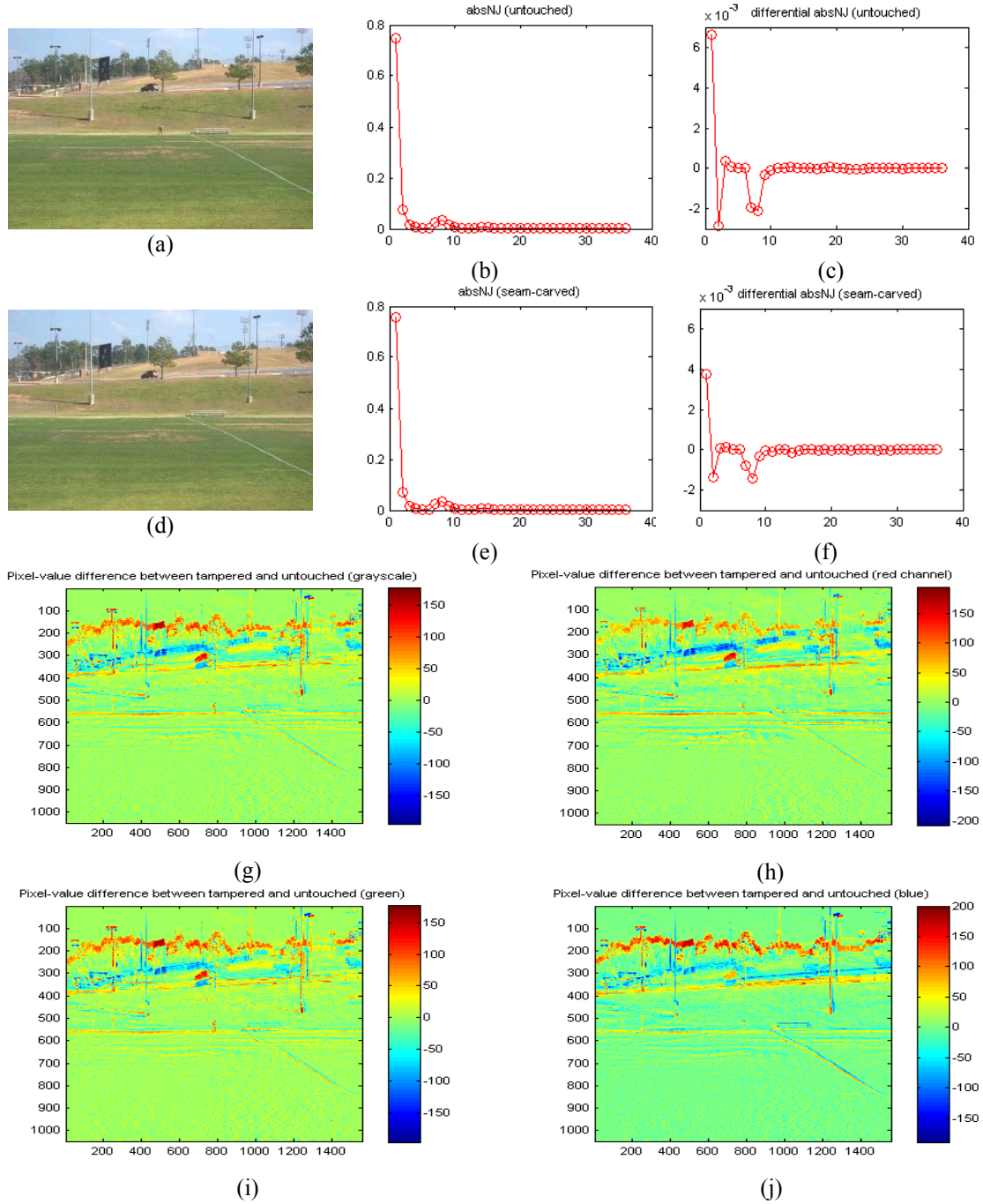


Figure II-6. Untouched JPEG image (a) and the forged image (d). Original neighboring joint densities in DCT domain are shown in (b) and (e), and the differential densities between original density and calibrated density are given in (c), and (f), respectively. The differences of the pixel values between the tampered (d) and untouched (a) are shown in (g) on the grayscale, (h) on the red channel, (i) on the green channel, and (j) on the blue channel.

*II-3-b. Experimental design*

We adopted 500 JPEG images with a standard quantization table of the quality '75'. The seam carving forgery tool at the website http://code.google.com/p/seam-carving-gui/ is used to modify JPEG images. The small objects are removed from the images at first in the spatial domain by using the tool, and the doctored images are stored in JPEG at the same quality of the untouched image. This avoids double JPEG compression for possible exposure by the detection of double JPEG compression. Figure II-7 shows several untouched images (on the left) and tampered images (on the right) by seam-carving in our experiment.

While data embedding in JPEG-based steganography directly modifies quantized DCT coefficients in transform domain, seam carving inserts or removes seams with minimum cost from one end of the image and modifies the pixel values directly in the spatial domain. The modification generally destructs original JPEG compression block, resulting in the change of the joint density in DCT domain. To detect seam carved forgery in JPEG images, in addition to the approach of calibrated neighboring joint density features in DCT domain, we also make use of SRMQ1, a detector of spatial domain rich models originally designed to detect spatial-domain-based steganography (Fridrich and Kodovsky 2012). We surmise that SRMQ1 may capture the statistical modification in spatial domain that was caused by seam-carving; therefore, we integrate CC-absNJ with SRMQ1 to detect seam carved tampering in JPEG images. Meanwhile, we conjecture that most steganalysis detectors are also effective in detecting this manipulation.

Figure II-7. Image samples in our experiments. The untouched is shown on the left and modified on the right. The objects highlighted by red circles on the left were removed by seam carving.

Due to the fact that for this experiment our forgery database is relatively small, we significantly increase the number of experiments for classification. We perform the experiment for each detector 2000 times with fisher linear discriminant and 1000 times with ensemble classifier. Generally, the computational cost by applying ensemble classifier to the detectors of rich models is much higher than fisher linear discriminant. In each case, 50% untouched images and 50% doctored images are randomly selected for training, and the remainders are used for testing.

## II-4. Detection of JPEG Double Compression

*II-4-a. Algorithm design*

*GGD, MGGD and JPEG-double compression*

Generalized Gaussian distribution (GGD) is widely used in modeling probability density function (PDF) of a multimedia signal. It is very often applied to transform coefficients such as discrete cosine transform (DCT) or wavelet ones. Experiments show that adaptively varying two parameters of the generalized Gaussian distribution (GGD) (Ohm 2004; Sharifi and Leon-Garcia 1995) can achieve a good probability distribution function (PDF) approximation, for the marginal density of transform coefficients. The GGD model is given by

$$\rho(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left\{-\left(|x|/\alpha\right)^{\beta}\right\} \tag{II-14}$$

Where $\Gamma(\cdot)$ is the Gamma function, scale parameter $\alpha$ models the width of the PDF peak, and shape parameter $\beta$ models the shape of the distribution.

An 8×8 DCT block has 64 frequency coefficients, our study shows that the marginal density of DCT coefficients at each specific frequency approximately follows the GGD distribution and some manipulation, for instance, double JPEG compression, changes the density. Figure II-8 demonstrates a singly compressed JPEG image with quality factor '75' (a), doubly compressed JPEG images with the first compression quality factor '55' (b) and '90' (c) respectively, followed by the second compression quality factor '75', and the marginal densities at frequency coordinates (2,1), (2,2), and (1,3).



(a)    (b)    (c)



(d)    (e)    (f)
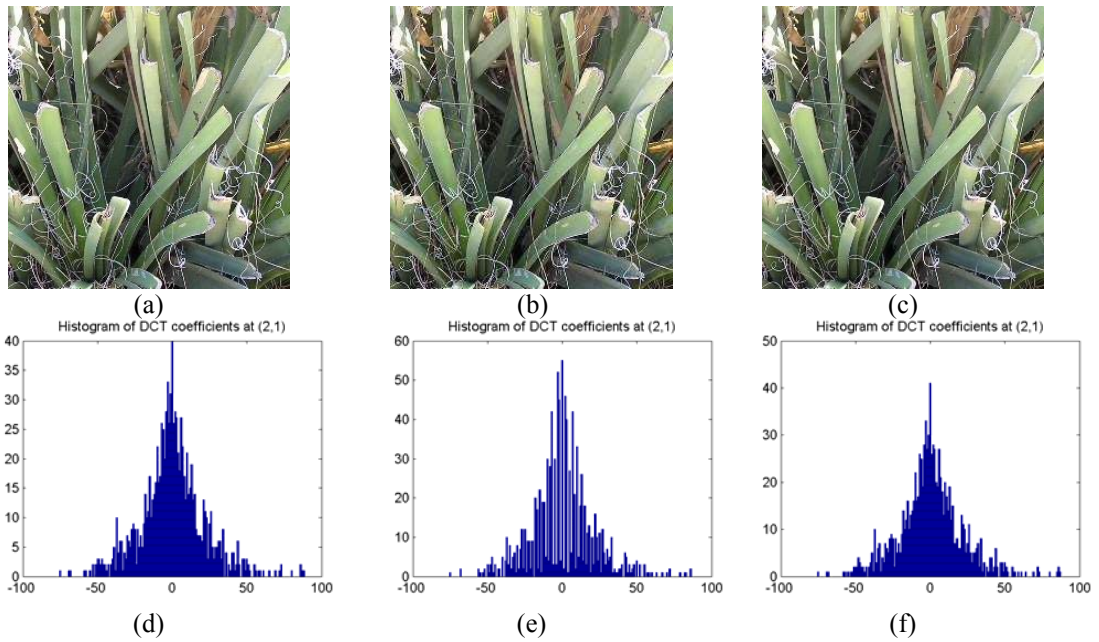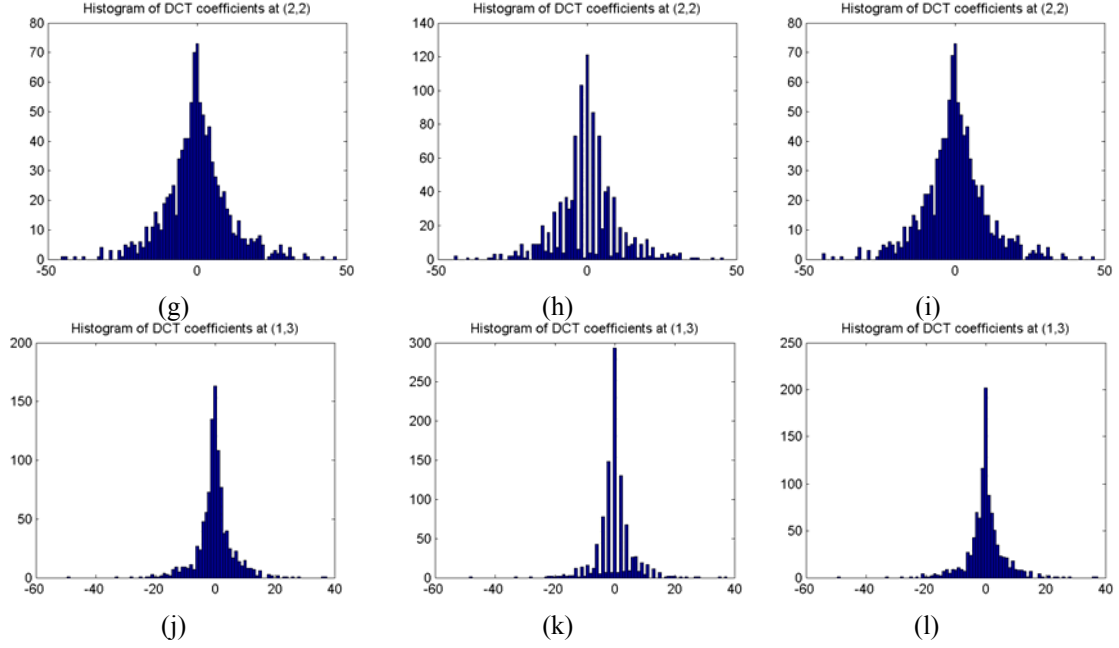
**Figure II-8.** Marginal densities of the singly compressed JPEG image (left) and the double compressions (middle and right). X-axis shows the values of the DCT coefficients and y-axis shows the occurrences.

Compared to the marginal density of the single compression, Figure II-8(d), (g), and (j), the modification caused by the double compression from the low quality factor '55', shown in Figure II-8(e), (h), and (k), is noticeable. However, the modification caused by the double compression from the high quality factor '90', Figure II-8(f), (i), and (l), is not big.

Although there does not appear to exist a generally agreed upon multivariate extension of the univariate generalized Gaussian distribution, some researchers define a parametric multivariate generalized Gaussian distribution (MGGD) model that closely fits the actual distribution of wavelet coefficients in clean natural images, exploit the dependency between the estimated wavelet coefficients and their neighbors or other coefficients in different subbands based on the extended GGD model, and achieve good image denoising (Cho and Bui 2005). The MDDG model is shown as follows:

$$p(\mathbf{x}) = \gamma \exp\left\{-\left(\frac{(\mathbf{x}-\mu)^t \Sigma_x^{-1}(\mathbf{x}-\mu)}{\alpha}\right)^\beta\right\} \tag{II-15}$$

Where $\gamma$ indicates a normalized constant defined by $\alpha$ and $\beta$, $\sum x$ is the covariance matrix and $\mu$ is the expectation vector.

To exploit the dependency between the compressed DCT coefficients and their neighbors, we study the neighboring joint density of the DCT coefficients, and postulate that some manipulation such as JPEG double compression will modify the neighboring joint density, shown by Figure II-9. Let the left (or upper) adjacent DCT coefficient be denoted by random vector X1 and the right (or lower) adjacent DCT coefficient be denoted by random

vector X2; let X = (X1, X2). The DCT neighboring joint density will be modified by the manipulation, and the change hence leaves a track for the manipulation. Figure II-9(a), (b), and (c) show the neighboring joint density of the singly compressed JPEG image of Figure II-8(a), of the doubly compressed JPEG image of Figure II-8(b), and of the doubly compressed JPEG image of Figure II-8(c). The differences of the neighboring joint density between the double compression and the single compression are given by Figure II-8(d) and (e). Figure II-9 verifies our postulation that the neighboring joint density has been modified by the double compression.



(a)  (b)  (c)

(d)  (e)
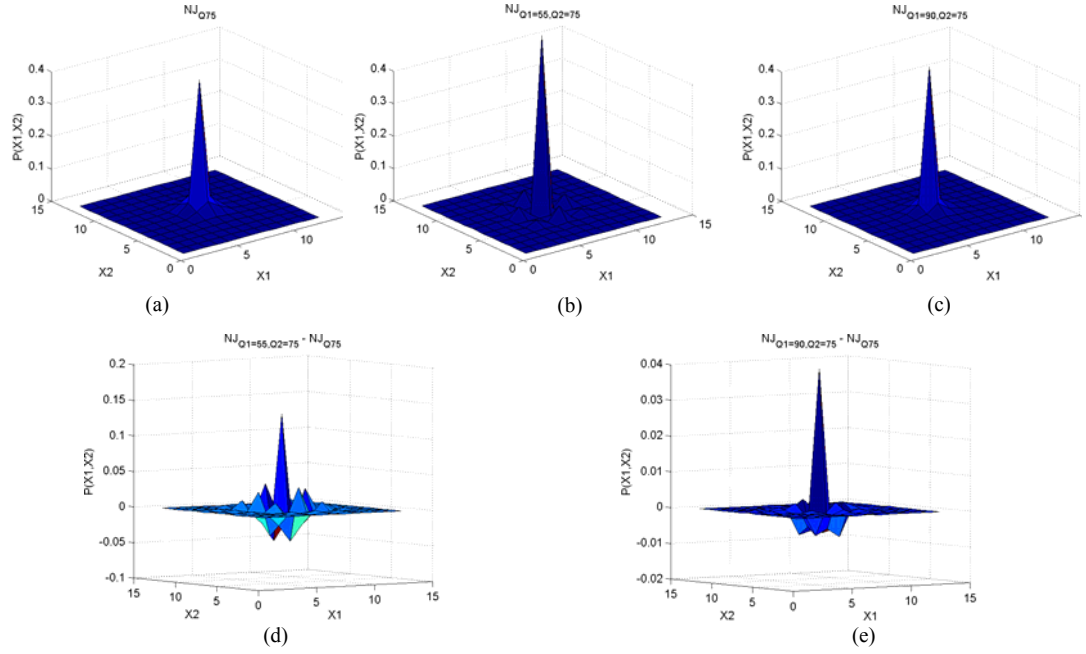
**Figure II-9**. Neighboring joint densities of the DCT arrays of the singly compressed JPEG image in Figure II-8(a) and the doubly compressed JPEG images in Figure II-8(b) and Figure II-8(c) and the differences.

*Feature Design*

Based on the statistical property and the observation of the modification caused by JPEG double compression, two types of features, marginal density and  neighboring joint density, are extracted and merged together as our detector. The details of feature mining is described as follows.

*Marginal Density Features*

Generally the manipulation to JPEG images will modify the DCT coefficients and change the marginal density of DCT coefficients at each specific frequency coordinate. In JPEG compression quantization table, the large values are aggregated in right bottom of the high-frequency coordinates and producing most zero-valued DCT coefficients after quantization at high frequency components. In other words, most non-zero DCT

coefficients are aggregated at low-frequency coordinates and the modification mostly occurs at low frequency subband, we design the following marginal density features at the low frequency of the absolute DCT coefficients.

An 8×8 DCT block consists of 64 frequency coefficients, with the frequency coordinates from (1, 1) to (8, 8), corresponding to the upper-left low frequency subband to the right-bottom high frequency subband. Let F denote the DCT coefficient array of a JPEG image, which consists of $M \times N$ blocks, $F_{ij}$ ($i$ =1, 2, …, $M$; $j$ = 1, 2, …, $N$). We select the low frequency coordinates

$$S = \{(2,1), (1, 2), (1, 3), (2, 2), (3,1), (1,4), (2,3), (3,2), (4,1)\}. \qquad \text{(II-16)}$$

The feature set consists of the following probability values

$$X = \left\{ \frac{1}{MN} \left( h_{kl}(0), h_{kl}(1), h_{kl}(2), h_{kl}(3), h_{kl}(4) \right) \middle| (k,l) \in S \right\}, \qquad \text{(II-17)}$$

where $h_{kl}(m)$ denotes the histogram of the absolute DCT coefficient at frequency coordinate $(k,l)$ with the value $m$ ($m$=0, 1, 2, 3, 4). Therefore, there are total 45 features in the marginal density set.

*Neighboring Joint Density Features*

The extraction of neighboring joint density features has been stated in the first part of Methods for JPEG Steganalysis.

In detecting JPEG double compression, we integrate marginal density features and neighboring joint density features together.

*II-4-b. Experimental design*

The original 5150 TIFF raw format digital images are obtained in 24-bit lossless true color and never compressed format. The single compressed images are generated by applying JPEG compression to these uncompressed images with different quality factors from 40, 45, 50, …, 90, the step size 5. The double JPEG compression is implemented by uncompressing the single compressed images and then compressed in JPEG format with different quality factors from 40, 45, 50, 55, …, 90, the step size is 5. The first and second JPEG compression quality factors are recorded as "Q1" and "Q2", respectively.

In our previous work on image and audio steganalysis, we have demonstrated that the image complexity is a significant parameter for the evaluation of steganalysis performance (Liu et al 2006; Liu and Sung 2007; Liu et al 2008a, 2008b; Liu et al. 2009a, 2009b). So far no work has been published to illustrate the relationship between detection performance on double JPEG compression and the image complexity, which will be

addressed in this study. Following our previous work in steganalysis, the shape parameter β of GGD of the DCT coefficients is used to measure the image complexity. All images are classified as five groups:

(a) β < 0.3, low image complexity
(b) 0.3 ≤ β < 0.4, low-middle image complexity
(c) 0.4 ≤ β < 0.5, middle image complexity
(d) 0.5 ≤ β <0.6, middle-high image complexity
(e) 0.6 ≤ β, high image complexity

We apply support vector machines (SVM) with RBF kernels (Vapnik 1998) to the feature sets extracted from these five groups for identification of double JPEG compression. Thirty experiments are run for testing each type of feature set in each group. Average testing accuracy is compared.

**II-5 Identification of Smartphone Image Source and Manipulation**

*II-5-a. Algorithm design*

Our detection method is the same to that used in detecting JPEG double compression.

*II-5-b. Experimental design*

We adopt five different types of smartphones from four manufacturers to capture images. Those images were taken randomly without any particular requirement. The information on these smartphone images are listed in Table II-2 and some image samples are shown in Figure II-10.

**Table II-2.** Original images obtained by smartphones

| Smartphone brand | # Images | Format |
|---|---|---|
| HTC G3 | 149 | JPEG |
| HTC HD2 | 114 | JPEG |
| Huawei U8150 | 141 | JPEG |
| Iphone 3 | 70 | JPEG |
| Nokia E71 | 125 | JPEG |
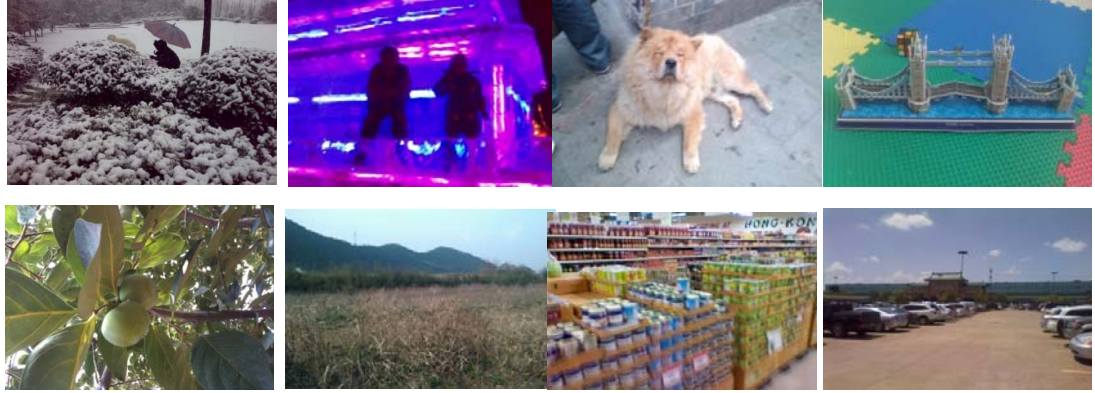
**Figure II-10**. A few sample images used in our experiment.

All these original images are manipulated by using the following six types of operations:

I.   All original images are trans-coded to the JPEG format with the standard quantization table at quality factor of '75'. In other words, these images are uncompressed first and then recompressed at quality factor of '75';

II.  The first four rows and first four columns are cut from original images in spatial domain and the remaining pixel values are trans-coded to the JPEG format with the standard quantization table at quality factor of '75'. In other words, these images are uncompressed first, followed by cropping, and then recompressed at standard quality factor of '75';

III. The first four rows and first four columns are cut from the original images, the remaining data are resized by multiplication with the scale factors of 0.7 and 2, respectively in spatial domain, and then trans-coded to the JPEG format with standard quantization table at quality factor of '75';

IV.  All original images are resized by multiplication of original image size with the scale factors of 0.3, 0.5, 0.8, 1.5, and 2, respectively in spatial domain, and trans-coded to the JPEG format with standard quantization table at quality factor of '75';

V.   The original images are trans-coded to the images with standard quantization table at quality of '75', cropped to remove the first four rows and first four columns in spatial domain, then resized by multiplication with the scale factors of 0.5 and 1.5, respectively in spatial domain, and finally trans-coded to JPEG format at quality of '75';

VI.  The original images are trans-coded to the images with standard quantization table at quality of '75', and resized by multiplication with the scale factors of 0.5 and 1.5, respectively in spatial domain, and then trans-coded to JPEG format at quality of '75'.

To sum up, these six types of operations include different scale parameters, which result in 13 series of operations as shown in the second column of Table II-3. These 13 operations are applied to each type of the total of 599 smartphone images and thus total of 7,787 processed images are generated. Since each of the operations are applied to the

five different smartphone brands, 65 class labels are generated in our experiments as listed in Table II-3.

Table II-3. The 65 class labels in our experiments

| Type | Scale factor | Class label | | | | |
|------|------|------|------|------|------|------|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| I | / | 1 | 14 | 27 | 40 | 53 |
| II | / | 2 | 15 | 28 | 41 | 54 |
| III | 0.7 | 3 | 16 | 29 | 42 | 55 |
| | 2 | 4 | 17 | 30 | 43 | 56 |
| IV | 1.5 | 5 | 18 | 31 | 44 | 57 |
| | 2 | 6 | 19 | 32 | 45 | 58 |
| | 0.3 | 7 | 20 | 33 | 46 | 59 |
| | 0.5 | 8 | 21 | 34 | 47 | 60 |
| | 0.8 | 9 | 22 | 35 | 48 | 61 |
| V | 0.5 | 10 | 23 | 36 | 49 | 62 |
| | 1.5 | 11 | 24 | 37 | 50 | 63 |
| VI | 0.5 | 12 | 25 | 38 | 51 | 64 |
| | 1.5 | 13 | 26 | 39 | 52 | 65 |

## II-6. Detection of MPEG Double Compression

*II-6-a. Algorithm Design*

In order to reveal the impact of the quantization and de-quantization process, we follow a double MPEG-2 compression process, analyze the difference between distributions of original DCT coefficients and reconstructed DCT coefficient, and finally explore the trace left by double compression.

*Distribution of reconstructed DCT coefficients*

After a test sequence compressed by a MEPG-2 encoder, two DCT terms in the DCT coefficient matrix, i.e., DCT (1, 2) and DCT (2, 1), are selected to create the corresponding statistical model of reconstructed DCT coefficients. At the remaining frequency terms, some may be influenced by the rounding error in de-quantization process, and the others do not have sufficient statistical quantity of non-zero quantized DCT coefficients due to large quantization step sizes. As a result, it is difficult to analyze and obtain robust statistical features from these frequency terms. Using the absolute value of DCT coefficients at those two DCT terms, Figure II-11(a) and (b) respectively illustrate the histogram of original non-zero DCT coefficients, $H_O(n)$, and the histogram of reconstructed non-zero DCT coefficients, $H_R(n)$.
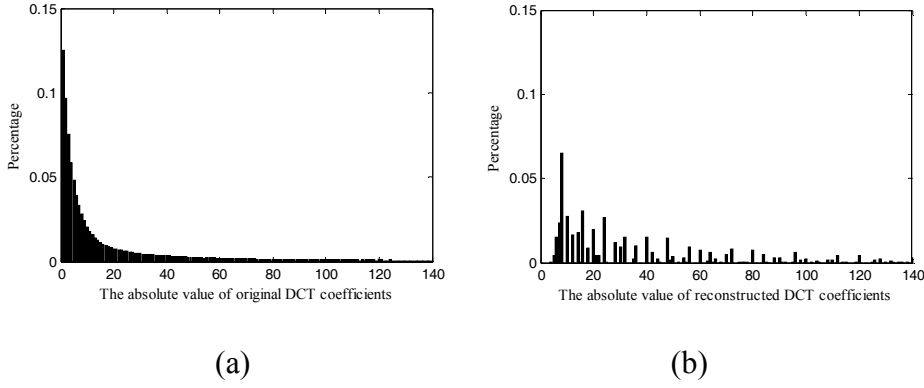
Figure II-11. Histograms of non-zero DCT coefficients at DCT (1, 2) and DCT (2, 1) in the intra frames: (a) $H_O(n)$, histogram of original DCT coefficients; (b) $H_R(n)$, histogram of reconstructed DCT coefficients.

Comparing these two histograms, $H_O(n)$ approximately follows the Laplacian distribution as traditionally described in the references (Reininger and Gibson 1983; Smooth and Lowe 1996), and after MPEG-2 compression and decompression process, some changes can be easily found in $H_R(n)$. Three significant changes can be summarized as follows:

**Discontinuity**, because the values of de-quantized DCT coefficients are only multiples of quantization scale factors, and some quantization scale factors, such as 9, 11, and 13, are not present in TM5, the number of some DCT coefficients in the $H_R(n)$ are close to zero.

**Peak Offsetting**, in the interval $(0, q_B]$ of $H_R(n)$, the maximum number of non-zero DCT coefficients does not occur at $n=1$ like Figure II-11(a), but at $n=q_B$. Because the number of reconstructed DCT coefficients is determined by two factors: the number of original DCT coefficients in the specific quantization interval and the probability distribution of quantization scale factors in the whole stream. Since the number of blocks quantized with quantization scale factor $q_B$ is much greater than that with other ones, the maximum value of $H_R(n)$ is shifted to the right.

**Approximate Periodicity**, the distribution of non-zero reconstructed DCT coefficients presents an approximate periodic distribution. Because most of quantization scale factors focus on the $q_B$, $H_R(m)$ is larger than its neighbor values when the coefficient value $m$ is the multiple of $q_B$. This feature is similar to the distribution of de-quantized DCT coefficients in JPEG images.

When reconstructed DCT coefficients from the primary compression, as input DCT coefficients, are entered into another MPEG-2 encoder, above statistic characteristics will lead to some artifacts into the distribution of doubly quantized DCT coefficients.

*Artifacts in the distribution of doubly quantized DCT coefficients*

In the decoding process, the quantization scale factor of each macro block can be read from video streams, so all macro blocks in the same type of frame can be divided into some subsets according to their quantization scale factors. In each subset, we can construct the distribution of quantized DCT coefficients at DCT (1, 2) and DCT (2, 1). Throughout this paper, we only consider the absolute value of DCT coefficients. Finally a set of distribution curves, $H(k, q, n)$, with different types of frame $k$ and quantization scale factor $q$ will be obtained. This process can be described as the diagram in Figure II-12.
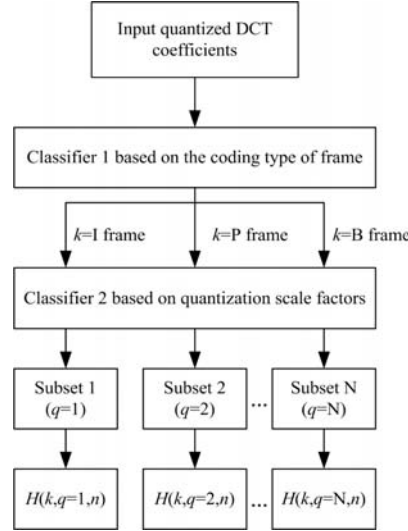
Figure II-12. The construction procedure of MEPG-2 compression detection model

If the choose of quantization scale factor of each macroblock is assumed to be independent of the DCT coefficients of that macroblock, the DCT distribution of each subset is similar to the input distribution. In the secondary compression, since the distribution of input DCT coefficients has been changed by the primary compression, the distribution of quantized DCT coefficients $H(k, q, n)$ presents some different properties to that of original compression, which can be utilized to determine the existence of double compression. In the following analysis, DCT coefficients of I frames are utilized to illustrate the abnormal phenomenon, and $H(k=I, q, n)$ is written as $H(q, n)$ for concision. In Figure II-13, the test sequence is firstly compressed by TM5 at 6 Mbps, then decoded and doubly compressed by a MPEG-2 encoder under different conditions. To concisely depict artifacts of double compression, we only present $H(q, n)$ when $q$ is even and not bigger than 8, and if the number of quantization scale factors equaling to $q$ is zero, the curve $H(q, n)$ will not be plotted. Fig. 5(a) indicates that the distribution curves of singly quantized DCT coefficients $H_6(q, n)$ can be approximately considered as a set of concave functions which are monotone decreasing in the interval [1, 6]. In the following interval, the curves of $H_6(q, n)$ maybe slightly fluctuate, because the number of large quantized DCT coefficients is relatively small, and easily affected by video content and other factors.
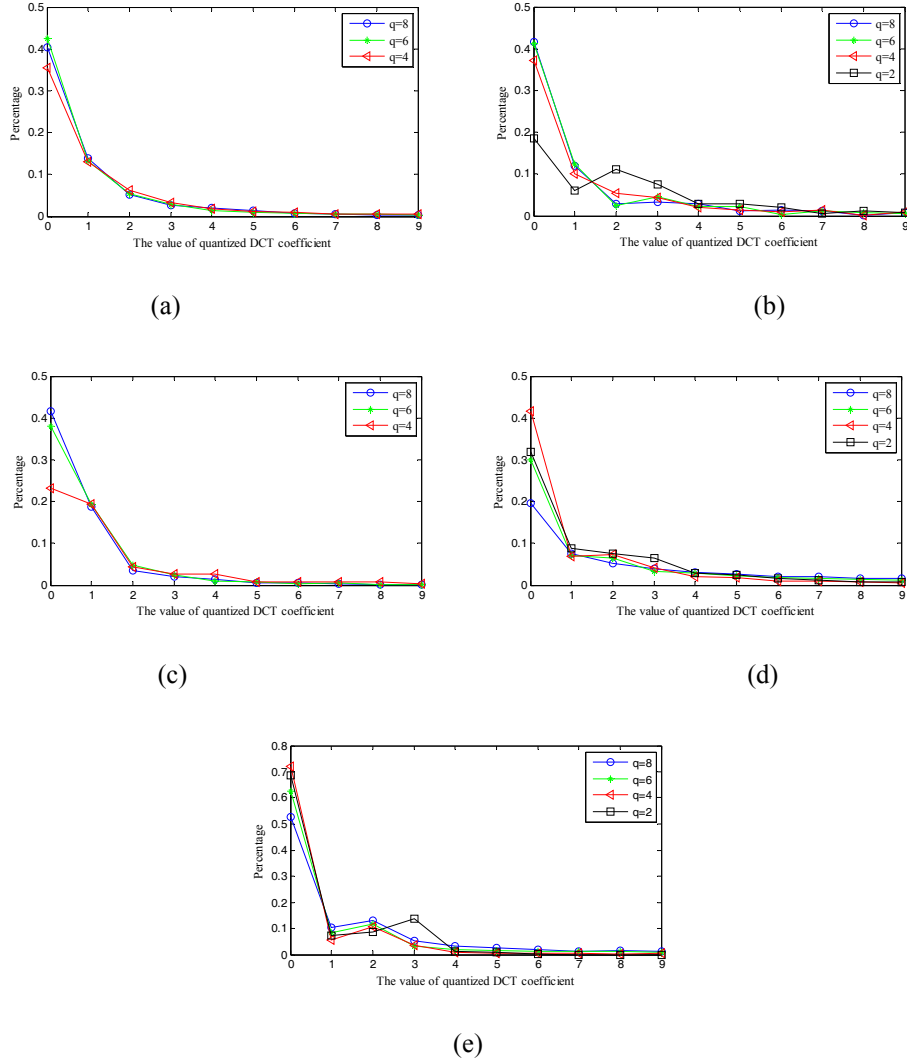
Figure II-13. Examples of double compression artifacts in histograms of quantized DCT coefficients under different conditions. (a) $H_6(q, n)$, singly compressed by TM5 at the constant bit-rate 6 Mbps; (b) $H_{6-7}(q, n)$, doubly compressed by TM5 at the constant bit-rate 6 Mbps followed by 7 Mbps; (c) $H_{6-5}(q, n)$, doubly compressed by TM5 at the constant bit-rate 6 Mbps followed by 5 Mbps; (d) $H_{6-6}(q, n)$, firstly compressed by TM5 at the constant bit-rate 6 Mbps and doubly compressed by Adobe Premiere 2.0 at the constant bit-rate 6 Mbps; (e) $H_{6-7}(P, q, n)$, the double compression artifacts in the histograms of quantized DCT coefficients in P frames which are doubly compressed by TM5 at the constant bit-rate 6 Mbps followed by 7 Mbps.

Notes: $H_X(q, n)$ and $H_{X-Y}(q, n)$ denote distribution curves of quantized DCT coefficients of single and double compression, respectively, and $X$, $Y$ denote the corresponding target output bit-rates (Mbps).

In Figure II-13(b), the MPEG-2 test sequence is re-compressed by TM5 at 7 Mbps, and its distribution of doubly quantized DCT coefficients is shown as $H_{6-7}(q, n)$ in which an obvious convex pattern appears at the $H_{6-7}(2, 2)$. As mentioned above that the input DCT distribution of each subset in the secondary compression is similar to the entire

distribution of input DCT coefficients which have been influenced by the primary compression, after the second quantization process, a convex pattern will arise at $H_{6\text{-}7}$ (2, $n$) because of the effect of *Peak Offsetting*, when the quantization scale factor of the secondary compression $q$=2 is much smaller than $q_B$ of the primary compression. On the other hand, in the interval [3, 6], part of values of $H_{6\text{-}7}(q, n)$ also do not strictly abide by a monotonic decline function. When the quantization scale factor $q$ of the secondary compression is not equal to $q_B$ of the primary compression, the effects of *Discontinuity* and *Approximate Periodicity* will lead to some convex patterns in the doubly quantized DCT distribution. However, these artifacts do not show obvious periodicities, because $H(q, n)$ decays very quickly.

In Figure II-13(c), the decoded test sequence is doubly encoded by TM5 at 5 Mbps, obtaining a set of histograms $H_{6\text{-}5}(q, n)$. Because the target output bit-rate of the secondary compression is smaller than that of the primary compression, the average value of quantization scale factors increases (as shown in Figure II-159), and some small quantization scale factors will disappear, such as $q$=2, but some convex patterns caused by double compression can also be detected in the other curves, such as $H_{6\text{-}5}(4, n)$.

In Figure II-13(d), the decoded test sequence is doubly encoded by Adobe Premiere 2.0 at 6 Mbps. Although the test video is doubly compressed at the same target output bit-rate, quantization scale factors in the secondary compression are independent of these in the primary compression. We can also detect convex patterns in the distribution curves of quantized DCT coefficients to verify the existence of double compression, such as $H_{6\text{-}6}(2, 3)$, $H_{6\text{-}6}(4, 2)$, and so on.

To sum up, in the set of distribution curves of quantized DCT coefficients in intra frames, the convex pattern can be viewed as a distinctive feature of the double MPEG compression under different output bit-rate conditions. Meanwhile, the statistic result also shows that those features can be observed in the inter frames (*P* frames or *B* frames), depicted in Figure II-13(e). As a result, we can design an effective double MPEG compression detection algorithm based on convex patterns.

*DOUBLE MPEG-2 COMPRESSION DETECTION SCHEME*

Based on the above statistical analysis, a new detection scheme is proposed for double MPEG-2 compression. Some features will be extracted from distributions of quantized DCT coefficients, and utilized to build a double MPEG-2 compression detector combined with a support vector machine (SVM) classifier.

When a MPEG-2 video stream is input into our detector, each GOP are defined as a sample. All DCT coefficient blocks in the same type of frames are firstly divided into some subsets according to quantization scale factors. Then in each subset, quantized DCT coefficients at two special DCT terms, DCT (1, 2) and DCT (2, 1), are assembled to construct a set of histogram curves $H(k, q, n)$, where $k$, $q$, and $n$ are the type of frame, the quantization scale factor, and the value of quantized DCT coefficients, respectively. In order to calculate the convex pattern, a detection function $T(k, q)$ is defined as (II-18) for

every curve $H(k, q, n)$.

$$T(k,q) = \sum_{n=1}^{N} \alpha_n \times t(k,q,n);$$ (II-18)

$$t(k,q,n) = \max\left( 0, \ 1 - \frac{H(k,q,n-1) + H(k,q,n+1)}{2 \times H(k,q,n)} \right).$$ (II-19)

Where $\alpha_n$ are a set of weighting factors to reflect the importance of the convex pattern at different positions. Since quantized DCT coefficients concentrate in a small range (as shown in Figure II-13), $N$ is set to 6 in our experiments. To illustrate the effectiveness of our extracted features, two standard test sequences are doubly compressed by TM5 with target output bit-rate 6 Mbps followed by 7 Mbps, and a feature vector consisting of $T$(I, 2), $T$(I, 4), and $T$(I, 6) is extracted to construct a 3D scatter chart, as shown in Figure II-14. In these two scatter charts, most samples of originally compressed streams are close to zero, and the separate clustering for the two cases is clear which makes classification possible.



(a)            (b)
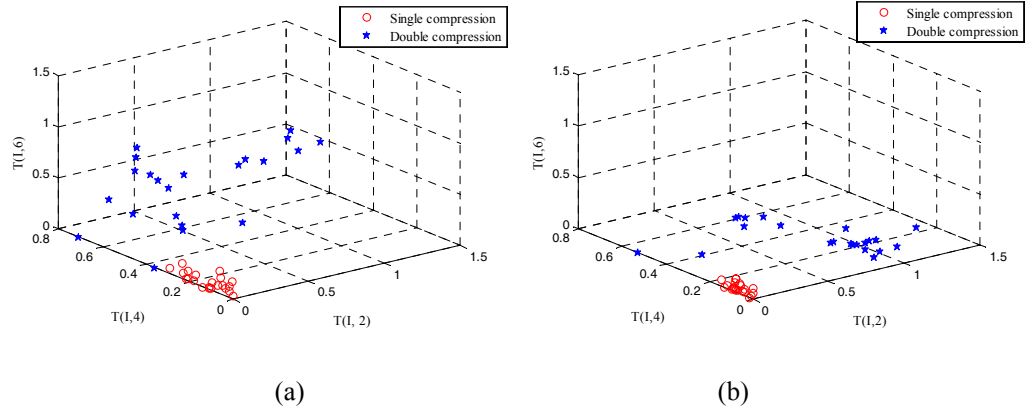
Figure II-14. Scatter charts of feature vectors ($T$(I, 2), $T$(I, 4), and $T$(I, 6)) where circles and stars denote single compression and double compression, respectively: (a) '*waterfall*' video sequence; (b) '*galleon*' video sequence

(a)                                                          (b)



(c)
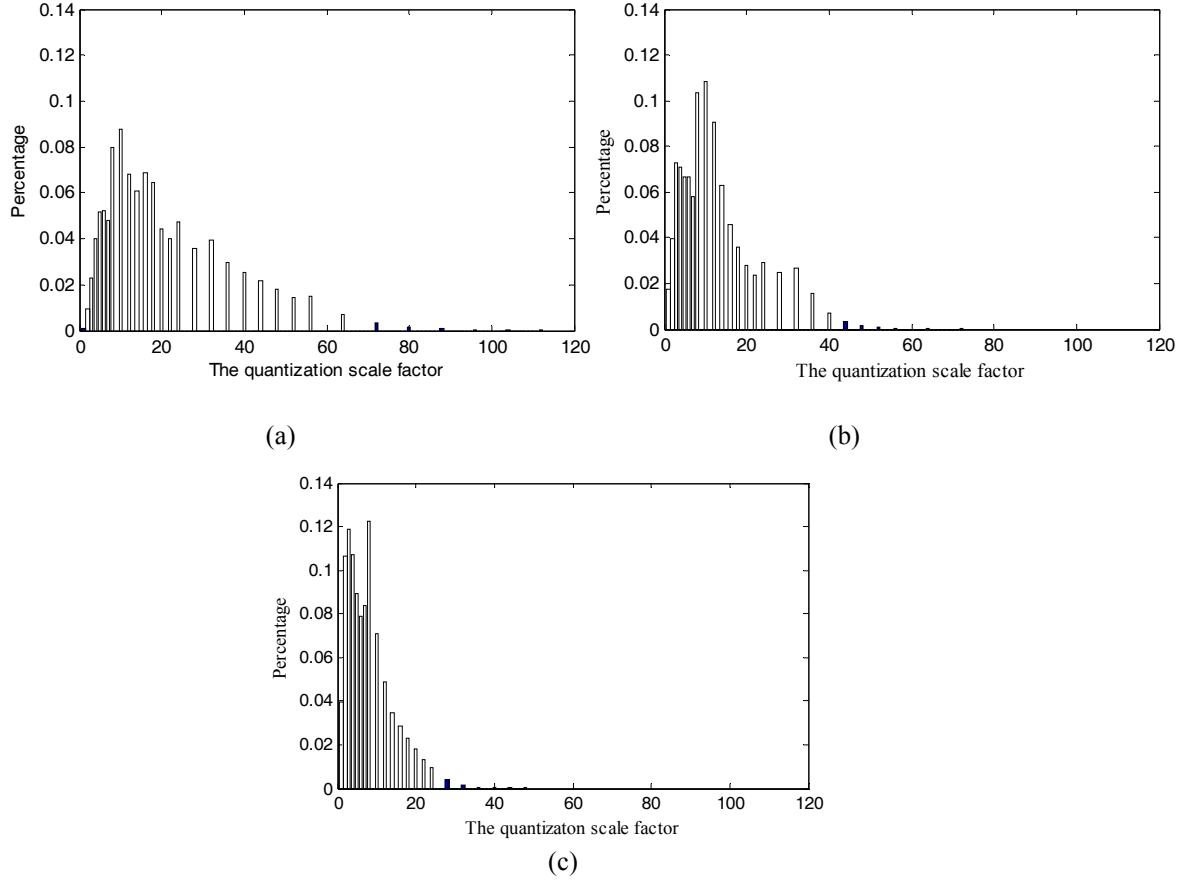
Figure II-15.  Histograms of quantization scale factors at different output bit-rates: (a) 4 Mbps; (b) 6 Mbps; (c) 8 Mbps.

The higher the target output bit rate is, the smaller the quantization scale factor tends to be, as show in Figure II-15. In order to approximately indicate the relationship between the quantization scale factor and the target output bit rate, all standard video sequences are originally compressed by TM5 at 4 Mbps, 6 Mbps, and 8 Mbps, respectively, and quantization scale factors of all video sequences at a certain constant output bit rate are assembled to construct the histogram. At low target output bit-rate, the percentage of small quantization scale factor is too small to stably and reliably present the convex pattern. To obtain a robust and effective scheme, the feature vector $V$ will be constructed as follows.

$$V = \begin{cases} \{T(k,q), q = 2,3,4,5,6,7,8; k \in \{I, P, B\}\}, & \text{if } BR \geq 6Mbps \\ \{T(k,q), q = 3,4,5,6,7,8,10; k \in \{I, P, B\}\}, & \text{if } 4Mbps < BR < 6Mbps \\ \{T(k,q), q = 4,5,6,7,8,10,12; k \in \{I, P, B\}\}, & \text{if } BR \leq 4Mbps \end{cases} \qquad \text{(II-20)}$$

In each type of frame, seven quantized DCT histograms are used to extract features and the total dimension of feature vector $V$ is 21 (7×3). Finally a widely used support vector machine tool, LIBSVM package (Chang and Lin 2011) is used to train or test classifiers. A radial basis function (*RBF*) is chosen as the kernel function and a grid search is performed to select the best parameters for the kernel. Each sample will be labeled by the classifier as being originally MPEG compressed or being doubly MPEG compressed.

49

After all samples in the testing video clips have been labeled, we count the number of samples classified as being doubly MPEG compressed. If the percentage of doubly MPEG compressed samples is larger than a threshold $T_p$, the current video clip will be classified as being doubly MPEG compressed.

*II-6-b. Experimental design*

Three sets of experiments are presented to show the validity and applicability of our proposed algorithm for detecting double compression. Four MPEG-2 encoders are introduced into our test system, including Test Model 5, the default MPEG-2 encoder in Adobe Premiere 2.0 (Web Adobe Premiere Pro), Sony HDR-XR500E, and Canon FS10E. The first two are MPEG-2 video coding software, and the latter two are hand-held digital video cameras. We select some different combinations of these MPEG-2 encoders to simulate the double compression process. The detection performance of the identification algorithm is measured in terms of recall and precision which are defined as follows.

$$Precision = \frac{C}{C+F}$$
(II-21)

$$Recall = \frac{C}{C+M}$$
(II-22)

Where $C$ represents the number of correctly detections of doubly MPEG-2 compressed videos, $F$ denotes the number of false alarms, and $M$ denotes the number of misses.

In our double MPEG-2 compression detector, according to a large number of experimental results, the default weighting parameters $\alpha_n$ in the detection function $T(k, q)$ are empirically set as: $\alpha_1 = 0.24, \alpha_2 = 0.24, \alpha_3 = 0.24, \ \alpha_4 = 0.1, \alpha_5 = 0.1, \alpha_6 = 0.08$. Each GOP of video clips is defined as a sample, and 21 features are extracted from each sample for training or testing. In all experiments, the ratio of training samples to test samples is 1:1, and the threshold $T_p$ is set as 0.5 with a simple majority voting rule.

a)      *Double MPEG-2 compression with the same MPEG-2 encoder*

In this experiment, the primary and secondary compression process are implemented with the same video coding software, Test Model 5 (abbr. TM5) or the default MPEG-2 encoder in Adobe Premiere 2.0 (abbr. Premiere). The target output bit-rate of the primary compression is set as 6 Mbps, while that of the secondary compression varies from 4 Mbps to 8 Mbps in steps of 1 Mbps except 6 Mbps. In the practical video application, these three bit rates (8 Mbps, 6 Mbps, and 4 Mbps) are usually utilized to generate the highest, the standard, and the worst video quality for the standard resolution format, respectively. We establish a video sequence dataset that consists of 50 test sequences, including 20 standard test sequences (220 frames each sequence) which come from Video Quality Experts Group (VQEG) (Web VQEG), and the other video clips (300 frames each clip) come from high definition DVD. The contents and motion complexity of the test sequences vary in a

large range. Some main parameters in these two encoders are described as Table II-4, and the other parameters can be set as default values.

Table II-4.  Predominant Parameters in Two MPEG-2 Software Encoders

| Parameters | Premiere Setting | TM5 Setting |
|---|---|---|
| TV standard | PAL(720×576) | PAL(720×576) |
| Frame Rate(f/s) | 25 | 25 |
| Frame Number | 220/300 | 220/300 |
| Pixel Aspect Ratio | 4:3 | 4:3 |
| Profile | Main | Main |
| Level | Main | Main |
| Bit-Rate(Mbps) | 4/5/6/7/8 | 4/5/6/7/8 |
| GOP Setting | M=3, N=12 | M=3, N=12 |
| VBV Buffer Size(kbits) | 112×16 | 112×16 |

*b)      Double MPEG-2 compression with different video encoders*

In this experiment, two DVs (digital video camcorders), Sony HDR-XR500E and Canon FS10E, are utilized to obtain the original MPEG-2 videos. We use each camcorder to record 50 nature video clips with length of 300 frames in our campus. In each test group, 50 nature image sequences are firstly encoded into MPEG-2 compressed files by the built-in encoder in the DV. These original compressed streams are input into PC, decoded and re-coded into 50 double compressed streams by the MPEG-2 video coding software -- Adobe Premiere Pro 2.0 or TM5.  Finally 200 test streams (100 single compressed streams and 100 double compressed streams) will be put into our detector to test its performance.

In the parameter settings, the most important parameters in DVs are resolution and target output bit-rate. The standard definition video format is selected as our encoding mode, whose resolution is 720×576 and output bit-rate is 6 Mbps. The other parameters just only affect the subjective effects of video resources, but have less impact on the statistical characteristics of DCT coefficient distribution, and we initial them as default values in the DVs. In the MPEG-2 software coders, all parameters are set the same as the MPEG-2 encoders in Section V.A, as shown in Table 1. We only adjust the target output bit-rate of software encoders to create new video files with different quality to test the adaptability of our detection scheme.

*c)      Double compression with frame tampering operation*

Frame tampering is one of the common video forgery operations, which can change the video content and confuse the viewers by removing some special frames in the video resources, such as some surveillance videos. In this experiment, 50 original compressed videos recorded by Sony HDR-XR500E at 6 Mbps are decoded into an image sequences, and the first three images are removed to simulate the frame tampering operation. Finally these doctored sequences are re-coded at different target output bit-rates by TM5, and 100 test streams (50 singly compressed streams and 50 doubly compressed streams) will be put into our detector to verify its robustness.

## II-7 MP3 Audio Steganalysis

### II-7-a. Algorithm design

In MP3, each frame consists of two granules, and each granule represents 576 16-bit PCM samples in a time sequence. Through compression, each frame is first divided into 32 adjacent frequency subbands and then converted into 576 finer subbands in the MDCT domain. From observations, we notice that the information-hiding behavior modifies most of the quantized MDCT coefficients in one frame at the same time with the exception of the coefficients with small absolute values, indicating that the intra-frame distribution is preserved. On the other hand, the inter-frame pattern is altered across adjacent frames. Based on this analysis, we designed inter-frame feature sets by utilizing second-order derivative-based spectrum analysis.

### Statistical Model and Signal Complexity

In image processing, several statistical models (Do and Vetterli 2002; Huang and Mumford 1999; Sharifi and Leon-Garcia 1995; Srivastava et al 2003; Winkler 1996) were introduced to illustrate the distribution of the intensity values of pixels, such as Markov random field models (MRFs), Gaussian mixture models (GMM), and generalized Gaussian density (GGD) models in transform domains. Experiments show that a good probability distribution function (PDF) approximation for the marginal density of coefficients at a particular subband produced by various types of wavelet transforms may be achieved by adaptively varying parameters of the GGD (Do and Vetterli 2002; Huang and Mumford 1999; Sharifi and Leon-Garcia 1995; Srivastava et al 2003; Wouwer et al 1999). The GGD model contains the Gaussian and Laplacian PDFs as special cases, using $\beta = 2$ and 1, respectively.

For MP3 digital audio, the GGD model also provides a faithful approximation of the distribution of quantized MDCT coefficients which varies with compression ratio and signal complexity. Therefore, as a useful measure of signal complexity, the shape parameter of the GGD becomes another important evaluation factor, in addition to embedding strength, for MP3 steganalysis. Figure II-16 illustrates some signal samples with different values of complexity measurement $\beta$. At the same embedding strength, we surmise that the signals with lower complexity are easier to be steganalyzed, but the steganalysis of the audio streams in high complexity is much harder, because the features become less discriminable in more complex signals.
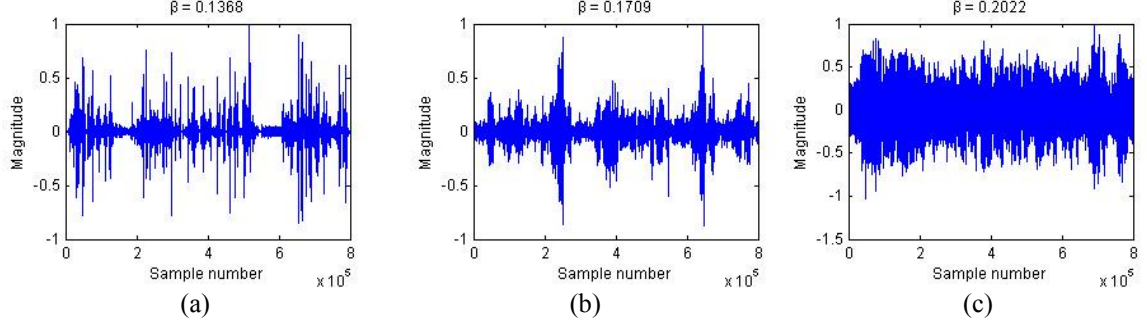
Figure II-16. MP3 audio signal samples with different values of signal complexity, β.

For long window, there are 576 MDCT coefficients in one frame. For short window, three consecutive groups of 192 coefficients are combined into one frame. For an audio signal with N frames, we define the quantized MDCT coefficients as a matrix:

$$IX = \begin{pmatrix} ix_{0,0} & \cdots & ix_{0,575} \\ \vdots & \ddots & \vdots \\ ix_{N-1,0} & \cdots & ix_{N-1,575} \end{pmatrix} \qquad \text{(II-23)}$$

In matrix *IX*, each row, denoted by *MDCT_F*, contains all quantized MDCT coefficients in one frame, and each column, denoted by *MDCT_B*, includes all quantized MDCT coefficients in one subband.

$$MDCT\_F_t = \begin{bmatrix} ix_{t,0} \cdots ix_{t,i} \cdots ix_{t,575} \end{bmatrix} \qquad \text{(II-24)}$$

$$MDCT\_B_i = \begin{bmatrix} ix_{0,i} \cdots ix_{t,i} \cdots ix_{N-1,i} \end{bmatrix}^T \qquad \text{(II-25)}$$

The GGD model of quantized MDCT coefficients of one MP3 audio is depicted in the following equation:

$$p(ix;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|ix|/\alpha)^\beta} \qquad \text{(II-26)}$$

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, z > 0 \qquad \text{(II-27)}$$

where $\Gamma(z)$ is the Gamma function and α models the width of the PDF peak (standard deviation), while $\beta$ is inversely proportional to the decreasing rate of the peak. Sometimes $\alpha$ is referred to as the scale parameter, and $\beta$ as the shape parameter. Generally, with the same compression ratio, the signal with complex variation has a high shape parameter of the GGD in the compression domain.

*Moment Statistics of GGD Shape Parameter*

MP3Stego embeds data into MP3 audio by randomly modifying the length of the data segment in frame headers. This information-hiding behavior increases the step-size of the quantization, resulting in a slight degradation of the quality of the audio. Across spectra, the absolute values of quantized MDCT coefficients decrease in some randomly selected frames. For selected frames, the magnitudes of all MDCT coefficients are decreased simultaneously.

Based on spectrum distribution analysis, we hypothesized that information-hiding behavior alters the continuity of the distributions of adjacent frames. Therefore, we designed a moment statistical analysis method on the shape parameter of GGD on inter-frame. The GGD distribution of an individual frame is modeled as:

$$p(ix_{t,i}; \alpha_t, \beta_t) = \frac{\beta_t}{2\alpha_t \Gamma(1/\beta_t)} e^{-(|ix_{t,i}|/\alpha_t)^{\beta_t}}$$

$$t = 0,1,2,...,N-1; \ i = 0,1,2,...,575$$

(II-28)

where $t$ is the frame index and $\alpha_t$ and $\beta_t$ are the scale parameter and shape parameter of the $t^{th}$ frame, respectively. Four moment statistical features are extracted from the spectrum of the shape parameter of the GGD. The mean value, standard deviation, skewness, and kurtosis are denoted by $M_\beta$, $\Delta_\beta$, $SK_\beta$ and $KU_\beta$, and calculated by the following equations:

$$M_\beta = \frac{\sum_{t=0}^{N-1} \beta_t}{N}$$

(II-29)

$$\Delta_\beta = \sqrt{\frac{1}{N} \sum_{t=0}^{N-1} (\beta_t - M_\beta)^2}$$

(II-30)

$$SK_\beta = \frac{\frac{1}{n} \sum_{t=0}^{N-1} (\beta_t - M_\beta)^3}{\left(\frac{1}{n} \sum_{t=0}^{N-1} (\beta_t - M_\beta)^2\right)^{3/2}}$$

(II-31)

$$KU_\beta = \frac{\frac{1}{n} \sum_{t=0}^{N-1} (\beta_t - M_\beta)^4}{\left(\frac{1}{n} \sum_{t=0}^{N-1} (\beta_t - M_\beta)^2\right)^2} - 3$$

(II-32)

*Frequency-Based Subband Moment Statistics*

In image processing, second-order derivatives are widely employed for detecting isolated points, edges, etc. To our knowledge, most audio steganography systems modify the bits of audio that also alter the pattern of second-order derivatives. Since MP3Stego randomly modifies the quantization step-size, the second-order derivatives of subbands also gain additional noise from information-hiding.

Let $f(x)$ ($x = 0,1,...,N-2$) denote the MDCT coefficients of MP3 audio at a specific frequency subband. The second-order derivative is defined as follows:

$$D_f^2(x) \equiv \frac{d^2 f}{dt^2} = f(x+2) - 2 * f(x+1) + f(x)$$

$$x = 0 \sim N-3$$

(II-33)

The MDCT coefficients of a stego-signal are denoted by $s(x)$, which may be modeled by adding a noise or error signal $e(x)$ into the original coefficient $f(x)$.

$$s(x) = f(x) + e(x)$$

(II-34)

The second-order derivatives of $e(x)$ and $s(x)$ are denoted by $D_e^2(x)$ and $D_s^2(x)$, respectively. We obtain:

$$D_s^2(x) = D_f^2(x) + D_e^2(x)$$

(II-35)

At this point, we present the following procedure to extract the second-order derivative-based statistics of the signals:

(1)  Obtain the second-order derivatives $D_{ix}^2(t,i)$ from 576 MDCT subband signals $MDCT\_B(i)$ across all frames where $t = 0,1,2,...,N-1$ .and $i = 0,1,2,...,575$.

(2)  Calculate statistics, including mean value, standard deviation, skewness, and kurtosis of subband signals.

(3)  To reduce the number of features, the whole frequency zone is divided into Z zones or parts (Z is set to 32 in our experiments) from the lowest to the highest frequency. We then calculated the sums of the mean value, standard deviation, skewness, and kurtosis in each zone, denoted by $M_Z$, $\Delta_Z$, $SK_Z$ and $KU_Z$, where $Z = 0,1,...,31$.

$$M_Z = \sum_{i=Z*18+1}^{Z*19} \frac{\sum_{t=0}^{N-3} D_{ix}^2(t,i)}{N-2} \tag{II-36}$$

$$\Delta_Z = \sum_{i=Z*18+1}^{Z*19} \sqrt{\frac{1}{N-2}\sum_{t=0}^{N-3}(D_{ix}^2(t,i)-M_Z)^2} \tag{II-37}$$

$$SK_Z = \sum_{i=Z*18+1}^{Z*19} \frac{\dfrac{1}{N-2}\sum_{t=0}^{N-3}(D_{ix}^2(t,i)-M_Z)^3}{\left(\dfrac{1}{N-2}\sum_{t=0}^{N-3}(D_{ix}^2(t,i)-M_Z)^2\right)^{3/2}} \tag{II-38}$$

$$KU_Z = \sum_{i=Z*18+1}^{Z*19} \frac{\dfrac{1}{N-2}\sum_{t=0}^{N-3}(D_{ix}^2(t,i)-M_Z)^4}{\left(\dfrac{1}{N-2}\sum_{t=0}^{N-3}(D_{ix}^2(t,i)-M_Z)^2\right)^{2}} - 3 \tag{II-39}$$

*Accumulative Neighboring Joint Density and Markov Approach*

The Markov process is a widely used stochastic process. In image steganalysis, Shi et al. (2007) proposed a Markov-based approach to detect the information-hiding in JPEG images. Liu et al. (2010) expanded the Markov features to the inter-blocks of the DCT domain. Although the designs of JPEG and MP3 compressions have similarities, the information-hiding process in digital audio does not share the same pattern with image steganography. Based on our previous analysis, we designed an inter-frame Markov approach (IM) and inter-frame Neighboring Joint Density (INJ) for MP3 audio steganalysis, described in the following equations, where $\delta = 1$ if its arguments are satisfied, otherwise $\delta = 0$. Similar to the references (Shi et al. 2007; Liu et al. 2010), the range of $i$ and $j$ is [-4, 4]. In such a case we have two 9×9 feature matrices with each one consisting of 81 elements or features. Figure II-17 shows the Markov transition probabilities of a cover and the steganogram in (a) and (b), the neighboring joint densities of the cover and the steganogram in (d) and (e), and the differences of the transition probabilities and the differences of the neighboring joint densities between the cover and the steganogram, in (c) and (f).
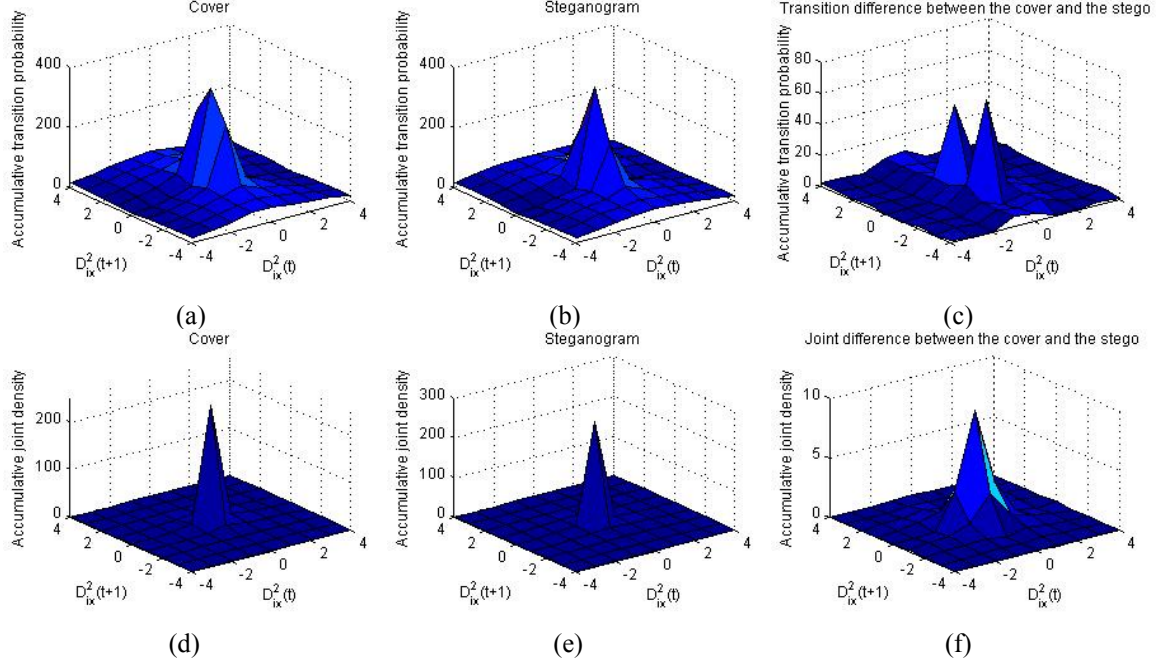
Figure II-17. The comparison of Markov transition probabilities of the second-order derivatives, shown in (a) and (b), the comparison of neighboring joint densities of the second-order derivatives, shown in (d) and (e), and the differences of the transition probability and neighboring joint density between a cover and the steganogram, shown in (c) and (f).

$$ IM(u,v) = \sum_{i=0}^{575} \frac{\sum_{t=0}^{N-4} \delta(D_{ix}^2(t,i)=u, D_{ix}^2(t+1,i)=v)}{\sum_{t=0}^{N-4} \delta(D_{ix}^2(t,i)=u)} \tag{II-40} $$

$$ INJ(u,v) = \frac{\sum_{i=0}^{575} \sum_{t=0}^{N-4} \delta(D_{ix}^2(t,i)=u, D_{ix}^2(t+1,i)=v)}{576*(N-3)} \tag{II-41} $$

*Feature Selection*

To achieve better performance in detection, we combined different feature sets as a comprehensive approach. However, with more features being included in the feature set, the increasing feature dimension and feature redundancy compromise the performance and the efficiency of steganalysis. Feature selection methods are designed to find an optimal feature subset by eliminating features with little discriminative information. Therefore, in a comprehensive approach, feature selection can be a useful solution to further enhance the accuracy as well as reduce the overhead.

Most widely used feature selection methods could be categorized into filter, wrapper, and embedded methods. Filter methods select feature subsets based on performance evaluation metrics extracted from feature set and work with no dependency on reference to machine learning algorithms. Filter methods are generally less expensive than wrapper and embedded methods. However, filter methods consider the features as independent individuals and ignore possible interactions among features. The combination of features does not guarantee an enhanced performance, according to the performance evaluation of individual features. Moreover, filter methods intend to select features which correspond to high evaluation scores, which might generate more redundant yet less informative feature subsets. Avcibas et al. (2003) presented a universal steganalysis based on image quality metrics and utilized the one-way analysis of variance (ANOVA) for choosing good measures. This feature selection belongs to the filtering approach, and the final feature set may not be optimal. Wrapper methods wrap around particular machine learning algorithms that can assess the selected feature subsets by estimating classification errors and then building the final classifiers. One of the well-known methods is the Support Vector Machine—Recursive Feature Elimination (SVM-RFE), which refines the optimal feature set by using the SVM in a wrapper approach (Guyon et al 2002). Embedded methods are built into adaptive systems while simultaneously processing feature selection with a classifier.

To deal with the issue of feature selection in MP3 audio steganalysis, we compared three feature selections: ANOVA, SVM-RFE, and a two-step approach incorporating ANOVA with SVM-RFE.

*II-7-b. Experimental design*

We select 5000 mono MP3 audio clips with a bit rate of 128 kbps and a sample rate of 44.1 KHz. Each audio signal has duration of 20 seconds, and the file size is 313 KB. These audio files include digital speeches and songs in several languages, such as English, Chinese, Japanese, Korean, and several types of music including jazz, rock, blue, and natural sounds. The payloads include voice, video, image, text, executable codes, random bits, etc., with each steganogram carrying a unique payload. By embedding different amounts of data, we constructed four sets of MP3 stego-audio with approximate modification densities of 8%, 12%, 16%, and 20%, which carry payloads of 30, 60, 90, and 120 Bytes. At a modification density of 20%, the MP3Stego reaches its maximum hiding capacity. Cover MP3 audio was compressed by using the same MP3 encoder in the MP3Stego. In this study, we used modification density, defined as the proportion of the number of modified non-zero MDCT coefficients to the number of all non-zeros MDCT coefficients, instead of the hiding ratio to evaluate detection performance.

Four groups of features are extracted from covers and steganograms. Sixty percent of the feature sets were employed for constructing the classification model, while the other forty percent of the feature sets were used for testing. For every experimental setting, we conducted the experiment 100 times, with the training and the testing sets randomly chosen every time. The classification returned results consisting of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The testing accuracy was

calculated by W*TP/(TP+FN) + (1-W)*TN/(TN+FP), where W is a weighting factor at the range of [0, 1]. Without losing generality, W was set to 0.5 in our experiments. Support vector machines (SVM) with RBF kernels were used for detection.

## II-8 AAC Audio Forgery Detection

*II-8-a. Algorithm design*

Several types of processing may exist to create the forgery in AAC audio streams. While the sources of AAC audio files used for forgery production are encoded at different bit rates, or the AAC source audio files are encoded at the same bit rate, but the doctored AAC audio file is encoded at a different bit rate, such kind of forgery undergo different compressions, or double AAC compression. Similar to the detection of double MP3 compression, we may reveal the forgery or manipulation by identifying the double AAC compression. While the source AAC audio files are encoded at the same bit rate, they are composited together in the time domain, and finally encoded at the same bit rate, the identification of such forgery by using double compression detection method is not effective.

As pointed out in the reference (Liu 2011b), JPEG compression generally generates block artifacts. Similar to JPEG compression, AAC audio compression also introduces block (frame) artifacts. While two AAC audio files are manipulated together, or some part is removed from an AAC audio file in temporal domain, and doctored audio data are re-encoded in AAC format at the same bit rate, the original block artifacts are generally undermined, in other words, original block switching structure will be reshuffled with a part of the neighbor blocks. By revealing such reshuffling manipulation, we may locate the doctored areas in the AAC audio forgery that was encoded at the same bit rate. And hence, we propose a shift-recompression-based differential analysis to detect the forgery in AAC audio streams with the same compression bit rate, described as follows.

### *Shift-Recompression-based Differential Analysis Algorithm*

i. *Decode the examined AAC audio stream to temporal domain, denoted by a matrix $S(i,j)$ ($i=0,1, 2, ..., M$; j indicates the number of channel of the audio signal);*

ii. *Shift the matrix $S(i,j)$ by t samples in the temporal domain, $t \in \{1, 2 ..., N-1\}$, here 2N stands for the number of samples in a frame/block. For a stationary signal, AAC uses a block size of 2048 samples ($N = 1024$). A shifted temporal WAV signal $S'(i, j, t)$ is produced. $S'(i, j, t) = S(i-t, j)$, $i= t, t+1, t+2, ..., M$;*

iii. *For t=1:1023*

    1) *Encode the shifted temporal signal $S'(i, j, t)$ to AAC format at the same bit rate;*

    2) *Decode the encoded audio signal from the above step to temporal domain, denoted by $S''(i, j, t)$;*

    3) *Calculate the difference $D(i, j, t)= S'(i, j, t) - S''(i, j, t)$;*

*4) Shift-recompression based reshuffle characteristic features (SRSC) are given by:*

$$SRSC(t) = \frac{\sum_{(i,j)}|D(i,j,t)|}{\sum_{(i,j)}|S'(i,j,t)|} \tag{II-42}$$

*Where t = 1, 2, ..., 1023. There are 1023 SRSC features for a stationary AAC audio file.*

While AAC audio stream are tampered in temporal domain and original frame structures are generally broken, by checking the SRSC feature under each different shift-recompression, we surmise that untouched SRSC features and tampered SRSC features are different, especially at the corresponding shift. As a result, the manipulation can be revealed.

*II-8-b. Experiment design*

*a)    Detection of Cropping and Recompression*

To verify our proposed shift-recompression-based differential analysis, we select 1000 never compressed WAV files; each file is in the length of 20 seconds. These WAV files are compressed in AAC format by using FAAC encoder, which is based on the original ISO MPEG reference code (Web Audiocoding). To simulate the shift-recompression of AAC audio forgery manipulation, AAC audio files are decoded into temporal domain and cropped by different samples at the beginning of the audio signals, then re-encoded in AAC format at the same bit rate. In our study, we tried to produce the cropping database at each possible cropping, or the number of samples removed is set from 1 to 1023, however, the time-consuming is too high to complete. Therefore, in our experiments, the numbers of the samples cropped are only set as 5, 50, 200, 400, 480, 512, 750, 900, and 1000, respectively. 1023 SRSC features are extracted from 1000 untouched AAC audio files and from the nine categories of doctored AAC audio files.

*b)    Detection of AAC Tampering*

In this type of experiments, we randomly select 200 AAC audio files, and remove a few audio samples in the middle, with the block switch offset by 100, 300, 500, 700, and 900 samples, then encode the doctored audio signals into AAC format at the same bit rate. There are total of 1000 doctored AAC audio files. We apply shift-recompression-based differential analysis to each audio file (including untouched and doctored audio files), each audio file is equally divided into six segments, as a result, 1200 untouched segments and 3000 touched segments are obtained. SRSC features are extracted from each segment, in order to discriminate the doctored audio files from untouched files, and identify the doctored areas.

# III. Results

## III-1 JPEG Steganalysis

### A. Statement of Results:

Tables III-1 (A) to (E) list the mean values of detection accuracy over 100 experiments to detect F5, steghide, MB1, MB2, and adaptive steganography in JPEG image respectively. In the results, by applying each learning classifier to the nine detectors, the best testing accuracy is highlighted in bold; by applying the three learning classifiers to the nine detectors, the best testing accuracy in bold are squared.

In detecting F5 steganography, calibrated neighboring joint density is generally superior to other eight detectors, shown by Table III-1(A). Most best testing accuracy is obtained by CC-absNJ with LibSVM. In detecting steghide steganography, CC-absNJ and the union of CC-JRM and SRMQ1 generally outperform other seven detectors, shown by Table III-1(B). In detecting MB1 and MB2 steganography, shown by Tables III-1(C) and (D), a calibrated neighboring joint density-based detector (CC-absNJ) obtains the best detection accuracy. In adaptive steganalysis, by using an ensemble classifier, the union of CC-JRM and SRMQ1 performs the best in detecting the steganograms at relative payload 0.1 bpac, with the testing accuracy of 85.7%. The application of an ensemble classifier to another rich model detector (CC-C300) cannot obtain optimal base learning classifiers; the detection is not available at relative payload 0.1 bpac. While detecting adaptive steganograms at 0.15 bpac to 0.35 bpac, CC-absNJ are comparable to the union of CC-JRM and SRMQ1, delivering better detection accuracy than other 7 detectors.

### B. Tables:

TABLE III-1. The mean detection accuracy (%) over 100 experiments with LibSVM (S), Fisher Linear Discriminant (F), and Ensemble classifier (E)

(A). F5 Steganalysis

| Detector | Relative payload 0.051 | | | 0.077 | | | 0.105 | | | 0.137 | | | 0.185 | | | 0.282 | | | 0.354 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E |
| CC-absNJ | 94.4 | 94.6 | 92.0 | 96.6 | 95.4 | 94.1 | 98.4 | 97.1 | 97.4 | 99.0 | 98.2 | 98.5 | 99.3 | 98.9 | 99.1 | 99.9 | 99.6 | 99.4 | 100 | 99.6 | 99.7 |
| absNJ | 91.9 | 91.0 | 86.7 | 93.9 | 91.1 | 89.7 | 96.5 | 92.9 | 95.0 | 97.6 | 94.7 | 96.0 | 97.9 | 95.8 | 97.0 | 99.2 | 98.7 | 97.2 | 99.8 | 99.7 | 99.6 |
| CC-PEV | 81.0 | 90.4 | 85.6 | 85.3 | 92.0 | 89.2 | 91.7 | 96.5 | 95.2 | 94.0 | 97.7 | 97.0 | 97.3 | 99.3 | 98.7 | 98.9 | 99.6 | 99.6 | 99.8 | 99.8 | 99.9 |
| PEV | 85.6 | 86.4 | 81.0 | 90.0 | 88.6 | 84.7 | 94.5 | 94.3 | 92.9 | 96.5 | 96.6 | 95.8 | 97.9 | 98.9 | 98.3 | 99.1 | 99.5 | 99.2 | 99.8 | 99.8 | 99.9 |
| Markov | 68.1 | 75.7 | 69.5 | 66.9 | 76.6 | 74.1 | 75.5 | 85.2 | 82.7 | 76.6 | 91.6 | 88.7 | 86.9 | 96.5 | 94.6 | 95.0 | 97.7 | 96.7 | 97.7 | 99.4 | 99.3 |
| CC-C300 | x | 81.4 | 74.9 | x | 90.3 | 86.2 | x | 94.8 | 93.7 | x | 96.8 | 96.2 | x | 98.8 | 98.7 | x | 99.2 | 99.0 | x | 99.7 | 99.8 |
| CF | x | 77.0 | 78.4 | x | 87.0 | 85.2 | x | 88.5 | 93.4 | x | 89.4 | 96.7 | x | 92.8 | 98.6 | x | 95.3 | 99.1 | x | 99.0 | 99.8 |
| CC-JRM | x | 79.3 | 87.7 | x | 91.6 | 92.1 | x | 93.6 | 95.9 | x | 95.0 | 97.6 | x | 96.9 | 98.0 | x | 98.8 | 99.4 | x | 99.8 | 99.8 |
| CC-JRM+SRMQ1 | x | 88.2 | 82.0 | x | 93.6 | 88.4 | x | 95.4 | 95.3 | x | 97.0 | 97.9 | x | 97.9 | 98.0 | x | 99.3 | 99.4 | x | 99.9 | 99.9 |

| Detector | Relative payload | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.021 | | | 0.029 | | | 0.036 | | | 0.044 | | | 0.055 | | | 0.073 | | | 0.114 | | |
| | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E |
| CC-absNJ | 92.1 | 92.5 | 87.4 | 95.4 | 95.9 | 93.3 | 98.1 | 97.4 | 96.9 | 99.2 | 98.3 | 98.7 | 99.7 | 99.2 | 99.4 | 99.9 | 99.7 | 99.9 | 100 | 99.9 | 100 |
| absNJ | 88.9 | 88.1 | 80.8 | 92.0 | 91.0 | 86.8 | 95.0 | 93.4 | 92.0 | 97.3 | 95.5 | 95.8 | 98.1 | 96.3 | 96.8 | 99.4 | 98.1 | 98.8 | 99.8 | 99.3 | 99.7 |
| CC-PEV | 82.4 | 89.5 | 83.7 | 83.9 | 93.0 | 89.2 | 90.1 | 96.8 | 94.5 | 94.2 | 98.7 | 97.5 | 96.7 | 99.3 | 98.4 | 99.1 | 99.7 | 99.5 | 99.7 | 99.8 | 99.8 |
| PEV | 82.4 | 82.6 | 74.4 | 85.5 | 85.7 | 80.9 | 90.5 | 92.1 | 89.0 | 95.3 | 96.4 | 94.5 | 97.5 | 97.9 | 96.7 | 99.3 | 99.5 | 99.1 | 99.7 | 99.7 | 99.7 |
| Markov | 72.9 | 83.3 | 77.4 | 75.0 | 85.9 | 81.9 | 84.1 | 91.7 | 89.3 | 89.6 | 96.0 | 94.0 | 93.7 | 97.4 | 96.2 | 97.9 | 99.0 | 98.5 | 99.2 | 99.4 | 99.3 |
| CC-C300 | x | 74.8 | 65.6 | x | 81.5 | 72.1 | x | 87.4 | 79.8 | x | 91.5 | 85.5 | x | 95.2 | 91.6 | x | 97.8 | 96.9 | x | 98.9 | 98.7 |
| CF | x | 78.2 | 66.6 | x | 83.3 | 70.1 | x | 88.0 | 76.3 | x | 91.6 | 83.9 | x | 93.4 | 90.8 | x | 95.8 | 96.3 | x | 94.9 | 99.0 |
| CC-JRM | x | 85.1 | 75.3 | x | 90.4 | 84.2 | x | 94.3 | 91.0 | x | 96.5 | 95.5 | x | 97.8 | 97.5 | x | 98.9 | 99.2 | x | 98.5 | 99.7 |
| CC-JRM+SRMQ1 | x | 85.9 | 92.4 | x | 91.2 | 96.8 | x | 95.2 | 98.8 | x | 97.3 | 99.5 | x | 98.5 | 99.7 | x | 99.4 | 99.8 | x | 99.2 | 99.9 |

| Detector | Relative payload | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.073 | | | 0.089 | | | 0.094 | | | 0.125 | | | 0.172 | | | 0.183 | | | 0.195 | | |
| | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E |
| CC-absNJ | 99.5 | 98.1 | 98.0 | 99.9 | 99.7 | 99.8 | 99.7 | 99.5 | 99.5 | 99.9 | 98.6 | 98.6 | 99.9 | 99.7 | 99.9 | 100 | 99.8 | 100 | 100 | 99.9 | 100 |
| absNJ | 95.8 | 94.8 | 91.5 | 97.4 | 95.3 | 95.9 | 97.9 | 96.5 | 96.9 | 98.4 | 94.6 | 96.8 | 99.8 | 98.9 | 99.7 | 99.8 | 99.5 | 99.8 | 99.9 | 99.4 | 99.9 |
| CC-PEV | 93.9 | 96.1 | 93.7 | 95.5 | 98.5 | 97.5 | 95.5 | 98.4 | 97.7 | 96.1 | 97.5 | 96.2 | 99.6 | 99.8 | 99.7 | 99.5 | 99.8 | 99.8 | 98.9 | 99.9 | 99.8 |
| PEV | 94.2 | 92.2 | 90.2 | 96.0 | 95.6 | 93.6 | 95.6 | 95.1 | 93.6 | 95.5 | 93.8 | 91.8 | 99.6 | 99.3 | 99.3 | 99.7 | 99.4 | 99.5 | 99.8 | 99.3 | 99.4 |
| Markov | 90.8 | 92.0 | 89.3 | 90.5 | 94.5 | 93.0 | 92.2 | 95.0 | 93.6 | 90.3 | 93.4 | 90.9 | 99.1 | 99.3 | 99.0 | 99.3 | 99.3 | 99.2 | 97.8 | 99.4 | 99.3 |
| CC-C300 | x | 74.6 | 73.8 | x | 87.7 | 61.3 | x | 83.3 | 54.4 | x | 77.8 | 67.0 | x | 96.5 | 87.5 | x | 94.9 | 77.4 | x | 90.7 | 66.6 |
| CF | x | 57.0 | 88.9 | x | 91.4 | 93.1 | x | 89.6 | 86.3 | x | 85.7 | 82.7 | x | 97.7 | 98.9 | x | 98.3 | 98.8 | x | 96.7 | 96.1 |
| CC-JRM | x | 60.6 | 91.2 | x | 96.2 | 97.5 | x | 95.1 | 97.2 | x | 92.7 | 96.2 | x | 99.4 | 99.8 | x | 99.7 | 99.9 | x | 99.4 | 99.8 |
| CC-JRM+SRMQ1 | x | 64.4 | 92.7 | x | 97.1 | 95.3 | x | 96.1 | 94.9 | x | 94.3 | 92.9 | x | 99.5 | 99.8 | x | 99.8 | 99.8 | x | 99.6 | 99.6 |

| Detector | Relative payload | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.101 | | | 0.120 | | | 0.131 | | | 0.168 | | | 0.226 | | | 0.245 | | | 0.271 | | |
| | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E |
| CC-absNJ | 98.5 | 96.4 | 95.5 | 99.3 | 98.4 | 98.4 | 99.7 | 99.1 | 99.4 | 99.8 | 99.1 | 99.2 | 100 | 99.7 | 99.8 | 100 | 99.9 | 99.9 | 100 | 99.9 | 99.9 |
| absNJ | 96.6 | 92.2 | 93.2 | 98.0 | 95.9 | 96.5 | 99.0 | 97.4 | 97.8 | 99.4 | 97.6 | 98.4 | 100 | 99.2 | 99.7 | 99.9 | 99.8 | 99.9 | 100 | 99.8 | 99.9 |
| CC-PEV | 95.0 | 96.7 | 95.4 | 95.5 | 98.9 | 98.1 | 97.0 | 99.3 | 99.1 | 98.9 | 99.5 | 99.2 | 99.7 | 99.9 | 99.9 | 99.6 | 99.9 | 99.9 | 99.8 | 99.9 | 99.9 |
| PEV | 94.0 | 92.3 | 90.6 | 96.2 | 95.8 | 94.4 | 98.0 | 97.8 | 97.4 | 99.2 | 98.9 | 98.6 | 99.9 | 99.8 | 99.7 | 99.8 | 99.7 | 99.7 | 99.9 | 99.9 | 99.9 |
| Markov | 90.7 | 92.0 | 89.9 | 87.2 | 94.7 | 92.6 | 92.4 | 96.5 | 95.2 | 96.4 | 97.1 | 96.1 | 98.3 | 99.2 | 98.9 | 98.5 | 99.5 | 99.3 | 99.3 | 99.7 | 99.7 |
| CC-C300 | x | 68.9 | 63.9 | x | 84.9 | 56.8 | x | 90.2 | 66.5 | x | 95.5 | 79.7 | x | 96.9 | 80.8 | x | 95.7 | 78.2 | x | 97.5 | 89.4 |
| CF | x | 56.8 | 83.9 | x | 89.3 | 86.2 | x | 92.6 | 92.1 | x | 93.4 | 96.4 | x | 97.3 | 98.0 | x | 98.0 | 98.4 | x | 99.0 | 99.2 |
| CC-JRM | x | 60.8 | 90.4 | x | 94.5 | 95.7 | x | 96.1 | 97.5 | x | 97.0 | 98.2 | x | 99.2 | 99.7 | x | 99.5 | 99.8 | x | 99.6 | 99.9 |
| CC-JRM+SRMQ1 | x | 63.3 | 90.4 | x | 95.4 | 94.5 | x | 97.1 | 96.6 | x | 97.8 | 98.1 | x | 99.4 | 99.6 | x | 99.6 | 99.8 | x | 99.7 | 99.9 |

(E). Adaptive Steganalysis

| Detector | Relative payload | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.1 | | | 0.15 | | | 0.2 | | | 0.25 | | | 0.3 | | | 0.35 | | |
| | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E | S | F | E |
| CC-absNJ | 77.8 | 78.0 | 78.3 | 89.9 | 89.5 | 89.9 | 95.7 | 95.1 | 95.4 | 98.6 | 97.6 | 98.1 | 99.3 | 98.5 | 99.0 | 99.6 | 99.0 | 99.5 |
| absNJ | 69.6 | 71.2 | 70.8 | 81.5 | 83.6 | 83.3 | 89.2 | 90.9 | 90.4 | 93.7 | 94.7 | 94.7 | 96.0 | 96.7 | 96.9 | 97.9 | 97.8 | 98.3 |
| CC-PEV | 58.0 | 66.6 | 70.0 | 68.8 | 82.0 | 83.1 | 76.5 | 90.6 | 90.9 | 84.4 | 96.0 | 96.0 | 89.5 | 97.6 | 97.8 | 94.7 | 99.0 | 98.9 |
| PEV | 66.0 | 64.5 | 65.6 | 77.7 | 78.0 | 78.6 | 86.3 | 87.7 | 87.6 | 92.9 | 94.2 | 94.0 | 95.8 | 96.7 | 96.7 | 98.1 | 98.8 | 98.7 |
| Markov | 50.1 | 51.5 | 50.9 | 53.3 | 66.6 | 67.1 | 57.5 | 77.8 | 78.8 | 65.8 | 85.7 | 87.4 | 69.4 | 91.4 | 92.4 | 73.5 | 94.8 | 95.3 |
| CC-C300 | x | 82.0 | NA* | x | 89.5 | 63.8 | x | 93.6 | 84.8 | x | 96.5 | 93.3 | x | 97.8 | 96.3 | x | 98.6 | 98.0 |
| CF | x | 81.0 | 83.6 | x | 87.6 | 90.5 | x | 91.9 | 94.5 | x | 95.1 | 97.0 | x | 96.3 | 98.0 | x | 97.9 | 98.9 |
| CC-JRM | x | 81.1 | 81.9 | x | 88.5 | 89.8 | x | 92.5 | 94.0 | x | 95.8 | 96.7 | x | 97.1 | 97.9 | x | 98.5 | 98.7 |
| CC-JRM+SRMQ1 | x | 83.8 | 85.7 | x | 91.1 | 88.6 | x | 94.8 | 95.7 | x | 97.0 | 98.3 | x | 98.1 | 99.2 | x | 99.2 | 99.6 |

* The testing results "NA" were caused by the failure of ensemble classifier while the final optimal base classifier may not be generated.

## III-2. YASS Steganalysis

### A.    Statement of Results:

In binary classification, the testing accuracy is measured by $0.5*TP/(TP+FN)+0.5*TN/(TN+FN)$. The mean testing accuracy over 200 experiments is given by Table III-2(A).  While the parameter noused is set to 19/14 while generating YASS steganograms, our method is generally more accurate han the other two compared methods. The zero-value density-based detection method (Li, Shi and Huang 2009) performs well when detecting the YASS steganograms that were produced with small B-block parameter; however, the detection performance apparently deteriorates while the parameter of B-block size increases. The experimental results are consistent with the results in the reference (Li, Shi and Huang 2009) and also validate our previous surmise.

In multiple-class classification, Tables III-2(B), (C), and (D) give the confusion matrix with the mean testing results over 200 times. While the parameter of *noused* is 19, detector of zero-value density hits the correct detection of 16.2% for covers, 84.4%, 73.4%, and 61.4% for YASS steganograms produced by large B-block size 13, 14, and 15 respectively. Our approach obtains correct detection results of 80.1%, 95.9%, 93.8%, and 90.3%, gaining considerable improvement.  While the parameter of noused is getting smaller, the detection performance of our detection method deceases. On average, our approach is better than zero-value density.

# B.    Tables:

TABLE III-2.  Mean Testing Accuracy (%) over 200 Experiments in Detecting YASS

### (A)    Binary-Class Classification

| *Noused** | B-block, T | diff-absNJ | | Zero-value density** | | CC-JRM+SRMQ1*** | |
|---|---|---|---|---|---|---|---|
| | | LibSVM | FLD | LibSVM | FLD | Ensemble | FLD |
| | 9 | 99.6 | 99.5 | **99.8** | 99.3 | 93.9 | 94.2 |
| | 10 | **99.3** | **99.3** | 98.8 | 98.9 | 95.7 | 94.5 |
| | 11 | 98.4 | **98.5** | 93.7 | 97.7 | 86.7 | 88.2 |
| 19 | 12 | 96.8 | **97.6** | 74.3 | 94.3 | 86.8 | 87.1 |
| | 13 | 96.0 | **96.4** | 61.5 | 86.6 | 75.3 | 79.9 |
| | 14 | 93.6 | **94.1** | 53.0 | 77.1 | 71.8 | 76.5 |
| | 15 | 89.6 | **90.1** | 48.3 | 69.9 | 63.3 | 71.7 |
| | 9 | 98.3 | 98.4 | **99.7** | 98.3 | 86.4 | 91.3 |
| | 10 | 97.3 | 97.9 | 95.2 | **98.3** | 90.4 | 91.3 |
| | 11 | 95.6 | 96.3 | 80.4 | **96.4** | 76.3 | 85.7 |
| 14 | 12 | 93.4 | **94.3** | 67.5 | 90.3 | 76.5 | 83.1 |
| | 13 | 90.7 | **91.5** | 62.3 | 84.7 | 62.0 | 76.1 |
| | 14 | 86.3 | **86.7** | 60.4 | 75.0 | 59.4 | 73.7 |
| | 15 | 81.7 | **82.1** | 58.8 | 68.5 | 51.7 | 72.3 |
| | 9 | 95.9 | 96.1 | **99.3** | 98.6 | 74.9 | 84.3 |
| | 10 | 93.6 | 93.7 | 91.8 | **98.6** | 74.8 | 82.6 |
| | 11 | 90.2 | 90.6 | 83.6 | **96.0** | 62.6 | 78.5 |
| 9 | 12 | 87.3 | 87.1 | 73.5 | **91.3** | 62.2 | 76.5 |
| | 13 | 82.3 | 82.7 | 69.5 | **86.7** | 50.5 | 71.4 |
| | 14 | 76.8 | 76.9 | 67.2 | **80.1** | NA | 67.9 |
| | 15 | 72.7 | 73.1 | 66.1 | **74.7** | NA | 66.0 |

\* *noused* is the parameter to set the number of the first few AC DCT coefficients for data embedding in the block.

\*\* zero-value density-based approach assumes prior knowledge of the exact embedding position of the first few AC DCT coefficients in zigzag order for data embedding, which is generally inapplicable and not assumed by our approach and rich models-based detection.

\*\*\* When applying an ensemble classifier (Kodovsky, Fridrich and Holub 201) to the rich model-based approach, the testing results "NA" means not available and is caused by the failure of the ensemble classifier while the final optimal base classifier may not be generated.

(B). Confusion Matrix of Mean Testing (%) over 200 Experiments with LibSVM in Detecting YASS
(Multiple-class classification, *noused* = 19)

| Truth | | Prediction accuracy, % | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Zero-value density | | | | | | | | Diff-absNJ | | | | | | | |
| | | cover | Steganogram | | | | | | | cover | steganogram | | | | | | |
| | | | T=9 | T=10 | T=11 | T=12 | T=13 | T=14 | T=15 | | T=9 | T=10 | T=11 | T=12 | T=13 | T=14 | T=15 |
| cover | | **16.2** | 0 | 0.9 | 5.1 | 8.4 | 18.2 | 23.9 | 27.3 | **80.1** | 0.0 | 0.1 | 0.5 | 1.7 | 2.4 | 5.4 | 9.9 |
| | T=9 | 0.0 | **100.0** | 0 | 0 | 0 | 0 | 0.0 | 0 | 0.1 | **99.8** | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 |
| | T=10 | 0.4 | 0 | **99.3** | 0 | 0 | 0.0 | 0.2 | 0.0 | 0.4 | 0 | **99.5** | 0.0 | 0 | 0.0 | 0.1 | 0.0 |
| | T=11 | 1.5 | 0 | 0 | **97.1** | 0.1 | 0.4 | 0.5 | 0.4 | 1.3 | 0.0 | 0.0 | **98.1** | 0.1 | 0.0 | 0.1 | 0.3 |
| steg | T=12 | 2.2 | 0 | 0.0 | 0.1 | **94.6** | 0.7 | 1.1 | 1.3 | 2.0 | 0.0 | 0.0 | 0.1 | **97.3** | 0.2 | 0.3 | 0.3 |
| | T=13 | 4.2 | 0 | 0.1 | 0.5 | 2.4 | **84.4** | 4.6 | 3.7 | 3.1 | 0.0 | 0.0 | 0.1 | 0.2 | **95.9** | 0.3 | 0.4 |
| | T=14 | 5.1 | 0 | 0.1 | 1.5 | 3.2 | 6.2 | **73.4** | 10.1 | 4.3 | 0.0 | 0.0 | 0.1 | 0.2 | 0.2 | **93.8** | 1.3 |
| | T=15 | 7.1 | 0 | 0.6 | 1.4 | 5.2 | 9.3 | 15.0 | **61.4** | 7.6 | 0.0 | 0.0 | 0.1 | 0.5 | 0.5 | 1.0 | **90.3** |

(C). Confusion Matrix of Mean Testing (%) over 200 Experiments with LibSVM in Detecting YASS
(Multiple-class classification, *noused* = 14)

| Truth | | Prediction accuracy, % | | | | | | | | | | | | | | | |
| | | Zero-value density | | | | | | | | Diff-absNJ | | | | | | | |
| | | cover | Steganogram | | | | | | | cover | steganogram | | | | | | |
| | | | T=9 | T=10 | T=11 | T=12 | T=13 | T=14 | T=15 | | T=9 | T=10 | T=11 | T=12 | T=13 | T=14 | T=15 |
| | cover | **17.1** | 0 | 1.3 | 6.6 | 13.0 | 17.9 | 21.3 | 23.0 | **52.9** | 0.4 | 1.1 | 2.0 | 4.6 | 7.4 | 14.0 | 18.4 |
| | T=9 | 0.0 | **99.9** | 0 | 0 | 0 | 0 | 0 | 0.1 | 0.7 | **98.4** | 0 | 0.1 | 0.2 | 0.2 | 0.2 | 0.3 |
| | T=10 | 0.9 | 0 | **96.9** | 0.0 | 0.3 | 0.2 | 0.8 | 0.8 | 1.2 | 0.0 | **97.2** | 0.1 | 0.2 | 0.2 | 0.5 | 0.6 |
| | T=11 | 1.4 | 0 | 0.2 | **93.2** | 0.6 | 1.2 | 1.8 | 1.7 | 2.1 | 0.6 | 0.2 | **95.2** | 0.4 | 0.1 | 0.7 | 1.1 |
| steg | T=12 | 2.3 | 0 | 0.6 | 0.5 | **87.9** | 2.5 | 3.1 | 3.2 | 3.6 | 0.2 | 0.2 | 0.4 | **92.1** | 0.9 | 1.1 | 1.4 |
| | T=13 | 3.2 | 0 | 0.8 | 1.7 | 4.3 | **75.9** | 8.1 | 6.0 | 4.9 | 0.1 | 0.2 | 0.3 | 1.0 | **89.8** | 1.8 | 1.9 |
| | T=14 | 5.5 | 0.0 | 1.8 | 3.9 | 6.0 | 12.1 | **59.9** | 10.9 | 6.9 | 1.8 | 0.6 | 0.4 | 0.6 | 1.7 | **85.1** | 4.5 |
| | T=15 | 7.4 | 0.1 | 1.7 | 4.8 | 8.1 | 15.0 | 20.3 | **42.7** | 9.8 | 0.3 | 0.4 | 0.7 | 1.1 | 2.4 | 3.9 | **81.5** |

(D). Confusion Matrix of Mean Testing (%) over 200 Experiments with LibSVM in Detecting YASS
(Multiple-class classification, *noused* = 9)

| Truth | | Prediction accuracy, % | | | | | | | | | | | | | | | |
| | | Zero-value density | | | | | | | | Diff-absNJ | | | | | | | |
| | | cover | Steganogram | | | | | | | cover | steganogram | | | | | | |
| | | | T=9 | T=10 | T=11 | T=12 | T=13 | T=14 | T=15 | | T=9 | T=10 | T=11 | T=12 | T=13 | T=14 | T=15 |
| | cover | **38.0** | 0 | 0.9 | 5.5 | 10.8 | 12.8 | 15.1 | 16.9 | **19.7** | 1.4 | 3.0 | 6.4 | 10.3 | 14.6 | 20.2 | 24.4 |
| | T=9 | 0.2 | **99.6** | 0 | 0 | 0.0 | 0 | 0.0 | 0.1 | 1.0 | **95.6** | 2.9 | 2.6 | 3.5 | 6.4 | 8.8 | 9.6 |
| | T=10 | 1.2 | 0.0 | **93.3** | 0.4 | 0.6 | 1.6 | 1.9 | 1.1 | 1.3 | 3.2 | **91.6** | 0.6 | 0.8 | 1.1 | 1.9 | 2.3 |
| | T=11 | 1.1 | 0 | 0.4 | **89.2** | 1.3 | 2.4 | 4.2 | 1.5 | 2.3 | 0.5 | 0.7 | **88.1** | 0.9 | 1.8 | 2.4 | 3.3 |
| steg | T=12 | 3.8 | 0 | 1.2 | 1.8 | **82.8** | 4.1 | 4.1 | 2.2 | 3.3 | 0.6 | 1.2 | 1.3 | **84.1** | 2.3 | 3.0 | 4.3 |
| | T=13 | 4.8 | 0.0 | 1.1 | 3.9 | 5.7 | **70.2** | 8.4 | 6.0 | 4.1 | 0.7 | 1.0 | 1.5 | 2.5 | **79.7** | 4.9 | 5.6 |
| | T=14 | 6.7 | 0.0 | 3.4 | 6.5 | 7.7 | 12.7 | **53.8** | 9.2 | 5.5 | 1.0 | 2.0 | 2.1 | 3.2 | 4.7 | **72.7** | 8.9 |
| | T=15 | 11.0 | 0.2 | 3.3 | 7.2 | 10.5 | 16.4 | 19.8 | **31.6** | 6.7 | 0.9 | 1.6 | 2.6 | 4.1 | 6.7 | 9.0 | **68.4** |

## III-3. Seam-carved Forgery Detection

### A.    Statement of Results:

The mean testing accuracy values are given in Table III-3 with 10 combinations of different detectors. While all these detectors were originally designed to detect JPEG-based steganography, all are effective to discriminate seam carved tampering from untouched. The union of calibrated neighboring joint density CC-absNJ with the detector of spatial domain rich model SRMQ1 obtains the best detection accuracy.

## B. Tables:

TABLE III-3. Seam-Carved Forgery Detection

Mean testing accuracy (%) over 2000 experiments with Fisher Linear Discriminant (**F**) and over 1000 experiments with Ensemble classifier (**E**)

| Detector | Mean detection accuracy, % | |
|---|---|---|
| | **F** | **E** |
| CC-absNJ | 94.8 | 94.8 |
| absNJ | 87.3 | 86.9 |
| CC-PEV | 85.6 | 92.7 |
| PEV | 87.7 | 87.8 |
| Markov | 88.2 | 92.0 |
| CC-C300 | 93.7 | 90.8 |
| CF | 94.2 | 95.3 |
| CC-JRM | 95.9 | 96.5 |
| SRMQ1 | 97.0 | 97.5 |
| CC-JRM+SRMQ1 | 96.8 | 97.1 |
| CC-absNJ+SRMQ1 | **97.2** | **97.6** |

## III-4. JPEG Double Compression Detection

### A. Statement of Results:

The average testing detection accuracy over 30 experiments in the five groups, are listed in Table III-4-1(A) to Table III-4-1(E), respectively, with the use of the feature sets: low-frequency histogram, Markov transition probability, expanded low-middle frequency histogram, and neighboring joint density, with the results shown in from the first row to the forth row, respectively. In each comparison, the highest average testing accuracy is highlighted in bold.

The results shows that expanded low-middle frequency histogram and neighboring joint density dominantly hit the bold values. Apparently, expanded low-middle frequency histogram approach is superior to the original low frequency histogram since it includes middle frequency histogram features. On average, neighboring joint density features outperform Markov transition probability features. The comparison among Table III-4-1(A) to (E) shows that image complexity plays a critically important role for the evaluation of detection performance. The detection accuracy in high image complexity, shown in Table III-4-1(E), is much less than other results, shown in Table III-4-1(A) to (D). It means that the identification of double JPEG compression in high image complexity is still challenging.

The identification of the first-time JPEG compression in double JPEG compression images is very useful for the further forensic analysis. To detect the first-time JPEG

compression or determine the quality factor of the first time JPEG compression in the double compressed images, we mix all double compressed image with the same Q2 factors but different Q1 factors, we merge the marginal features and neighboring joint density features together, and apply SVM to the features for multiple classifications. Table III-4-2 lists the testing accuracy to d distinguish the Q1 factors in all mixed images with the second quality factor '75'. Detection results show that the identification of the first-time JPEG compression in the mixed double compressed images is promising. However the detection in high image complexity is still challenging.

## B.    Tables:

Table III-4-1. The average detection accuracy over 30 experiments using low-frequency histogram feature set (results shown in the first row), Markov transition probability (results shown in the second row), expanded low-middle frequency histogram (results shown in the third row), and neighboring joint density features (results shown in fourth row).

(A) Detection accuracy % in low image complexity ($\beta < 0.3$)

| Q2 \ Q1 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **40** |  | 96.0 | 97.8 | 98.3 | 98.1 | 93.9 | 94.5 | 96.3 | 87.2 | 91.4 | 69.8 |
|  |  | 95.9 | 98.7 | 99.1 | 98.9 | 98.3 | 94.2 | 97.7 | 52.1 | 81.0 | 60.4 |
|  |  | 98.2 | 99.1 | 99.5 | 99.3 | 96.7 | **98.4** | 98.9 | **92.7** | **96.7** | **82.9** |
|  |  | **98.2** | **99.6** | **99.8** | **99.5** | **99.2** | 95.5 | **99.0** | 64.2 | 83.4 | 62.2 |
| **45** | 97.5 |  | **93.8** | 97.3 | 98.4 | 98.3 | 94.9 | 95.7 | 94.7 | 90.1 | 79.3 |
|  | 97.1 |  | 87.8 | 97.8 | 98.4 | 99.0 | 98.2 | 95.2 | 90.5 | 90.5 | 69.6 |
|  | 98.1 |  | 88.3 | 98.6 | **99.4** | 99.1 | 97.2 | **98.5** | **98.1** | **95.2** | **89.7** |
|  | **98.5** |  | 91.1 | **99.1** | 99.3 | **99.6** | **98.9** | 97.7 | 93.9 | 94.8 | 72.2 |
| **50** | 99.1 | **96.0** |  | **93.7** | 97.8 | 98.9 | 97.8 | 89.1 | 96.4 | 92.2 | 77.9 |
|  | 99.5 | 91.8 |  | 85.5 | 98.3 | 99.1 | 98.9 | 93.6 | 97.3 | 79.2 | 43.4 |
|  | 99.5 | 93.8 |  | 86.0 | 98.8 | 99.5 | 98.9 | 92.0 | 98.6 | **96.4** | **92.9** |
|  | **99.7** | 94.5 |  | 88.7 | **99.5** | **99.7** | **99.5** | **94.0** | **98.7** | 90.2 | 51.2 |
| **55** | 99.5 | 98.7 | 95.7 |  | **96.2** | 98.7 | 98.6 | 96.6 | 95.9 | 93.3 | 88.2 |
|  | 99.7 | 99.1 | 90.4 |  | 92.3 | 98.4 | 99.1 | 98.4 | 96.0 | 54.3 | 81.6 |
|  | 99.7 | 99.4 | 89.9 |  | 96.2 | 99.1 | 99.4 | 97.9 | 97.4 | **96.1** | **94.4** |
|  | **99.8** | **99.7** | **91.0** |  | 95.8 | **99.5** | **99.6** | **99.3** | **98.2** | 56.3 | 89.0 |
| **60** | **99.9** | 99.7 | 99.1 | 97.8 |  | 98.0 | 98.7 | 98.7 | 95.5 | 96.7 | 89.7 |
|  | 99.7 | 99.7 | 99.4 | 96.0 |  | 94.5 | 98.2 | 98.7 | 91.7 | 93.6 | 73.1 |
|  | 99.9 | **99.8** | 99.4 | 97.8 |  | **98.6** | **99.3** | 99.4 | **98.2** | **98.6** | **96.9** |
|  | **99.9** | **99.8** | **99.6** | **97.9** |  | 93.8 | **99.3** | **99.6** | 91.0 | 97.0 | 62.0 |

(B) Detection accuracy % in low-middle image complexity (0.3≤ β < 0.4)

| Q1 / Q2 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **40** | | 94.3<br>92.8<br>95.7<br>**96.1** | 96.3<br>97.5<br>98.2<br>**99.1** | 97.0<br>98.4<br>98.9<br>**99.4** | 97.0<br>98.0<br>**98.9**<br>98.9 | 92.4<br>96.9<br>94.3<br>**98.1** | 92.6<br>90.4<br>**97.4**<br>92.3 | 94.4<br>95.5<br>97.2<br>**97.9** | 82.7<br>46.5<br>**89.7**<br>58.1 | 88.2<br>73.0<br>**94.7**<br>77.6 | 61.5<br>53.4<br>**76.5**<br>54.9 |
| **45** | 96.2<br>95.5<br>96.6<br>**97.3** | | **90.5**<br>82.3<br>82.2<br>87.5 | 96.2<br>96.5<br>97.8<br>**98.1** | 97.4<br>97.5<br>**98.8**<br>98.6 | 97.1<br>98.3<br>98.8<br>**98.9** | 93.0<br>97.0<br>95.6<br>**97.9** | 94.6<br>92.4<br>**98.1**<br>95.4 | 91.4<br>86.3<br>**95.8**<br>90.7 | 86.4<br>85.6<br>**92.5**<br>90.8 | 73.4<br>63.1<br>**86.1**<br>65.4 |
| **50** | 98.7<br>99.1<br>99.4<br>**99.7** | **94.1**<br>88.3<br>91.2<br>91.8 | | 90.5<br>80.9<br>80.7<br>84.0 | 96.7<br>97.5<br>98.3<br>**98.8** | 97.9<br>98.7<br>99.1<br>**99.3** | 96.5<br>98.0<br>98.4<br>**98.8** | 86.5<br>90.4<br>89.2<br>**91.2** | 94.4<br>95.4<br>97.2<br>**97.6** | 89.3<br>75.3<br>**94.5**<br>85.9 | 72.8<br>41.2<br>**89.1**<br>48.5 |
| **55** | 99.6<br>99.6<br>99.7<br>**99.9** | 98.3<br>98.8<br>99.3<br>**99.7** | **93.5**<br>87.1<br>84.6<br>87.2 | | **94.4**<br>89.1<br>94.0<br>93.5 | 98.0<br>97.8<br>**99.0**<br>98.7 | 98.4<br>98.6<br>99.2<br>**99.4** | 95.0<br>97.5<br>97.1<br>**98.7** | 93.9<br>93.9<br>95.9<br>**96.8** | 90.7<br>49.3<br>**94.0**<br>52.1 | 85.6<br>77.6<br>**92.0**<br>84.3 |
| **60** | 99.8<br>99.6<br>**99.9**<br>**99.9** | 99.7<br>99.6<br>**99.8**<br>**99.9** | 98.8<br>99.1<br>**99.4**<br>**99.6** | **96.6**<br>93.7<br>96.5<br>96.0 | | 96.6<br>92.0<br>**97.8**<br>91.2 | 98.5<br>97.9<br>**99.3**<br>98.8 | 98.5<br>98.4<br>99.2<br>**99.3** | 94.2<br>88.9<br>**97.7**<br>87.1 | 95.5<br>91.3<br>**97.6**<br>94.6 | 87.8<br>68.6<br>**95.2**<br>58.8 |

(C) Detection accuracy % in middle image complexity (0.4 ≤ β < 0.5)

| Q1 / Q2 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **40** | | 93.1<br>91.5<br>94.5<br>**95.7** | 95.9<br>972<br>97.3<br>**98.5** | 96.6<br>97.9<br>98.4<br>**99.2** | 96.5<br>97.5<br>98.0<br>98.3 | 90.4<br>95.9<br>92.8<br>**97.1** | 90.4<br>87.8<br>**95.5**<br>89.7 | 92.8<br>94.1<br>95.4<br>**96.8** | 79.5<br>43.2<br>**87.1**<br>54.3 | 86.0<br>68.8<br>**92.6**<br>70.8 | 56.3<br>48.4<br>**71.7**<br>49.8 |
| **45** | 95.4<br>94.1<br>95.9<br>**96.3** | | **89.0**<br>80.1<br>78.4<br>84.5 | 95.0<br>95.6<br>96.7<br>**97.1** | 97.1<br>96.4<br>**98.2**<br>97.9 | 97.1<br>97.7<br>98.1<br>**98.7** | 92.0<br>95.8<br>94.3<br>**96.6** | 92.3<br>89.5<br>**96.2**<br>93.1 | 89.6<br>82.6<br>**94.0**<br>87.0 | 83.7<br>82.0<br>**90.8**<br>88.5 | 69.9<br>58.6<br>**83.5**<br>59.2 |
| **50** | 97.8<br>98.5<br>98.7<br>**99.1** | **93.5**<br>86.6<br>89.4<br>90.0 | | 89.8<br>78.8<br>77.1<br>82.6 | 96.2<br>96.6<br>97.1<br>**97.8** | 97.8<br>98.2<br>98.7<br>**99.1** | 96.6<br>97.5<br>97.7<br>**98.4** | 84.4<br>**89.6**<br>88.2<br>89.5 | 93.1<br>93.7<br>95.6<br>**96.2** | 87.6<br>70.2<br>**92.6**<br>82.8 | 68.9<br>40.4<br>**87.9**<br>47.5 |
| **55** | 98.7<br>99.1<br>99.3<br>**99.5** | 97.6<br>97.9<br>98.4<br>**99.1** | **93.2**<br>85.3<br>84.4<br>85.5 | | **93.5**<br>87.3<br>92.5<br>91.4 | 97.5<br>97.2<br>98.1<br>**98.2** | 97.9<br>98.1<br>98.7<br>**99.2** | 95.4<br>97.2<br>96.7<br>**98.2** | 92.6<br>92.2<br>94.6<br>**95.7** | 88.9<br>48.7<br>**92.7**<br>50.8 | 83.1<br>74.1<br>**91.1**<br>82.3 |
| **60** | 99.2<br>99.2<br>99.6<br>**99.7** | 99.0<br>99.0<br>99.3<br>**99.5** | 98.2<br>98.4<br>98.7<br>**99.0** | **96.2**<br>92.9<br>95.7<br>95.2 | | 96.5<br>91.5<br>**96.8**<br>90.6 | 97.9<br>97.0<br>**98.6**<br>98.4 | 98.3<br>98.1<br>98.9<br>**99.0** | 94.0<br>87.5<br>**96.9**<br>86.6 | 94.8<br>89.4<br>**96.7**<br>93.6 | 86.2<br>67.4<br>**94.1**<br>57.5 |

(D) Detection accuracy % in middle-high image complexity (0.5 ≤ β < 0.6)

| Q1 / Q2 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **40** |  | 87.6 | 91.8 | 94.1 | 94.6 | 86.7 | 84.3 | 86.3 | 65.5 | 74.7 | 46.8 |
|  |  | 85.5 | 94.8 | 96.1 | 95.5 | 92.3 | 80.1 | 89.3 | 38.6 | 54.5 | 39.7 |
|  |  | 89.3 | 93.7 | 96.3 | **96.3** | 87.2 | **91.4** | 90.9 | **74.3** | **86.5** | **55.9** |
|  |  | **89.7** | **96.5** | **98.2** | 96.2 | **94.8** | 82.5 | **93.5** | 49.1 | 56.1 | 45.0 |
| **45** | 92.1 |  | **80.0** | 91.0 | 94.1 | 94.6 | 88.1 | 86.5 | 82.1 | 71.8 | 53.6 |
|  | 89.8 |  | 63.8 | 91.8 | 93.3 | 95.1 | 92.2 | 82.6 | 71.8 | 71.0 | 44.9 |
|  | 92.0 |  | 62.4 | 93.0 | **95.9** | 96.3 | 90.5 | **91.4** | **89.3** | **81.3** | **67.4** |
|  | **93.5** |  | 72.0 | **94.2** | 95.5 | **96.7** | **94.2** | 87.8 | 78.2 | 79.2 | 48.4 |
| **50** | 96.5 | **89.4** |  | **82.4** | 92.2 | 95.9 | 94.4 | 79.3 | 87.6 | 79.1 | 54.0 |
|  | 97.7 | 77.7 |  | 65.5 | 94.1 | 97.0 | 95.6 | **85.5** | 89.4 | 56.9 | 37.7 |
|  | 96.8 | 81.8 |  | 62.0 | 94.0 | 97.3 | 96.1 | 84.7 | 91.6 | **86.0** | **76.8** |
|  | 97.7 | 83.6 |  | 74.7 | **95.8** | **97.6** | 96.6 | 84.1 | **92.7** | 70.3 | 45.8 |
| **55** | 98.0 | 95.8 | **89.6** |  | **89.0** | 95.3 | 96.4 | 93.3 | 88.1 | 83.1 | 71.5 |
|  | 99.0 | 97.1 | 77.8 |  | 79.8 | 95.2 | 97.2 | 95.0 | 87.1 | 43.8 | 60.5 |
|  | 98.0 | 96.5 | 74.4 |  | 87.1 | **96.8** | 97.6 | 94.1 | 88.5 | **87.6** | **84.7** |
|  | **99.1** | **98.1** | 78.9 |  | 86.9 | 96.2 | **97.9** | **96.5** | **91.3** | 47.9 | 67.9 |
| **60** | 99.4 | 98.5 | 96.6 | **93.3** |  | 93.3 | 96.4 | 97.2 | 91.4 | 91.2 | 78.8 |
|  | 99.1 | 98.9 | 97.5 | 88.6 |  | 87.9 | 95.3 | 97.2 | 85.0 | 83.9 | 56.2 |
|  | **99.5** | 98.2 | 96.8 | 91.1 |  | **94.3** | **97.1** | **98.0** | **95.5** | **94.0** | **90.5** |
|  | 99.4 | **99.2** | 97.8 | 92.4 |  | 85.4 | 96.6 | 97.8 | 80.9 | 88.9 | 53.7 |

(E) Detection accuracy % in high image complexity (0.6 ≤ β)

| Q1 / Q2 | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 75 | 80 | 85 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **40** |  | 59.7 | 69.9 | 77.8 | 79.5 | 73.0 | 65.6 | 61.3 | 43.8 | 49.6 | 41.9 |
|  |  | 59.7 | 75.3 | 80.4 | 81.5 | 79.0 | 57.8 | **66.4** | 38.1 | 43.7 | 38.2 |
|  |  | 58.0 | 70.3 | **81.0** | 81.9 | 74.2 | **70.7** | 63.3 | **48.5** | **55.5** | **46.7** |
|  |  | **63.6** | **77.6** | 80.1 | **82.0** | **80.3** | 48.3 | 62.2 | 41.9 | 43.2 | 42.2 |
| **45** | 68.6 |  | 50.9 | 67.6 | 74.3 | 81.5 | 75.2 | 72.6 | 51.8 | 51.4 | 45.3 |
|  | 69.0 |  | 44.8 | 68.0 | **76.8** | **82.1** | 80.2 | 57.7 | 50.7 | 47.9 | 41.7 |
|  | 68.9 |  | 49.8 | 67.3 | 76.3 | 81.7 | 77.0 | **76.0** | **55.7** | **53.7** | **48.3** |
|  | **71.3** |  | **51.8** | **69.5** | 75.8 | 80.9 | **80.2** | 54.8 | 47.8 | 51.5 | 45.5 |
| **50** | 77.8 | 60.5 |  | 49.2 | 72.0 | 82.2 | 83.1 | 65.5 | 57.6 | 50.6 | 43.2 |
|  | **84.1** | 47.4 |  | 44.5 | 75.7 | 83.2 | 83.3 | **70.8** | **66.4** | 43.8 | 38.1 |
|  | 81.2 | 56.6 |  | 48.6 | 72.2 | 82.8 | 81.9 | 67.6 | 64.4 | **54.3** | **47.7** |
|  | 82.7 | **61.4** |  | **53.9** | **76.4** | **83.6** | **84.3** | 68.5 | 64.3 | 48.2 | 43.6 |
| **55** | 86.0 | 80.2 | **62.2** |  | 60.5 | 80.9 | 85.1 | 83.0 | 61.5 | 51.8 | 47.8 |
|  | **90.6** | **82.7** | 48.6 |  | 50.0 | 79.5 | 84.0 | 81.7 | 64.3 | 40.7 | 42.6 |
|  | 85.4 | 79.2 | 51.0 |  | 57.8 | **81.3** | **85.5** | 80.4 | 63.8 | **52.3** | **51.0** |
|  | 84.5 | 80.3 | 56.1 |  | **63.0** | 77.9 | 85.2 | **84.6** | **66.7** | 44.9 | 46.5 |
| **60** | 89.7 | 84.4 | 79.3 | **70.1** |  | **73.6** | **83.1** | **85.9** | 72.9 | **64.2** | 50.2 |
|  | **93.2** | **86.9** | **86.8** | 62.4 |  | 61.9 | 79.4 | 83.6 | 65.9 | 58.4 | 41.1 |
|  | 90.6 | 84.8 | 80.4 | 66.1 |  | 69.1 | 82.5 | 85.3 | **78.9** | 64.2 | **58.1** |
|  | 90.8 | 85.5 | 78.2 | 69.5 |  | 59.4 | 77.2 | 84.2 | 62.5 | 56.4 | 45.9 |

Table III-4-2. Average detection accuracy (%) in each group to distinguish the first time JPEG compression factors by merging expanded low-middle frequency histogram and neighboring joint density features. The results in row 1 to row 5 conduct the detections in low image complexity to high image complexity.

| Q2 \ Q1 / β | | 40 | 45 | 50 | 55 | 60 | 65 | 70 | 80 | 85 | 90 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **75** | < 0.3 | 99.9 | 99.4 | 99.0 | 99.9 | 99.7 | 100.0 | 99.8 | 99.8 | 99.9 | 99.5 |
|  | [0.3,0.4) | 100.0 | 99.5 | 98.9 | 99.9 | 99.8 | 100 | 99.8 | 99.6 | 99.8 | 99.4 |
|  | [0.4, 0.5) | 100.0 | 98.8 | 97.8 | 99.6 | 99.6 | 99.9 | 99.4 | 99.7 | 99.8 | 98.9 |
|  | [0.5, 0.6) | 99.7 | 97.3 | 96.8 | 98.9 | 99.4 | 99.3 | 99.1 | 98.8 | 99.3 | 96.1 |
|  | > 0.6 | 91.9 | 74.0 | 79.0 | 86.0 | 90.6 | 87.9 | 70.8 | 78.3 | 89.2 | 50.0 |

69

### III-5. Detection of Smartphone Image Source and Post-Capture Processing

*III-5-a. Smartphone source identification*

#### A.     Statement of Results:

Experiment 1 — the operation to generate type I images is essentially JPEG double compression. Our first experiment is based only on type I images, to identify smartphone source. In each run, we randomly select 60% image samples from each brand of processed smartphone images, and the remaining 40% images are used for testing. LibSVM (Chang and Lin 2011) is employed for multiclass classification. We apply linear kernel and RBF kernel with default kernel parameters respectively and perform 100 runs with each kernel. The mean values of the confusion matrices on testing data over 100 experiments are summarized in Table III-5-1 by using our approach and 486-dimensional Markov approach (Chen and Shi 2008), respectively.

Experiment 2 — the operation to generate type II images is also essentially JPEG double compression, although cropping was first applied before double compression. Our second experiment is based on type I and type II images, to identify smartphone source. The experimental design and procedures are the same as those of Experiment 1. The mean values of the confusion matrices on testing data over 100 experiments are listed in Table III-5-2.

Experiment 3 — our third experiment is based on type III images with the scale factor value of 2, to identify smartphone source. The experimental design and procedures are the same as those of Experiment 1. The mean values of the confusion matrices on testing data over 100 experiments are listed in Table III-5-3.

Experiment 4 — this experiment is the same as those of Experiment 3 with the only difference that the type IV images with the scale factor value of 0.5 are used. The mean values of the confusion matrices on testing data over 100 experiments are listed in Table III-5-4.

Experiment 5 — this experiment is the same as those of Experiment 4 with the only difference that the type VI images with the scale factor value of 0.5 are used. The mean values of the confusion matrices on testing data over 100 experiments are listed in Table III-5-5.

Experiment 6 — all types of images with all scale factors are used in this experiment. Experimental design and procedure are the same as those of previous experiments. The mean values of the confusion matrices on testing data over 100 experiments are listed in Table III-5-6.

The experimental results show that it is promising to identify the original smartphone that was used to capture images, although the image under examination has been processed after capture. Overall, our approach outperforms Markov approach. By using Types I and II processed images, the accuracy of the smartphone source identification is apparently higher than the detection accuracy based on the other types. Type I only involves double JPEG compression. Type II was manipulated by cropping, followed by double-compression. It implies that double compression has major impacts on the DCT coefficients in Type II. In addition to double compression, images in other types were rescaled. In these types, double compression may or may not have major impacts to the modification of the property of DCT coefficients of the processed images, depending on the parameter value of each scale factor. Experimental results demonstrate that the detection accuracy on a large scale parameter (Table III-5-3) is much better than those on a small scale parameter (Tables III-5-4 and III-5-5). In our opinion, while the scale factor is small (<1), some information of original images will be lost, and hence the detection performance on the small scale factors deteriorates.

## B.    Tables:

Table III-5-1. Mean testing accuracy over 100 experiments based on type I images

### (a)  our approach

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **98.95** | 0 | 0.86 | 0.10 | 0.08 | **96.75** | 0 | 1.56 | 0.29 | 1.41 |
| | HTC HD2 | 0 | **100** | 0 | 0 | 0 | 0 | **98.67** | 0 | 0.98 | 0.36 |
| | Huawei U8150 | 0.09 | 0 | **99.82** | 0.09 | 0 | 1.64 | 0 | **97.82** | 0.52 | 0.02 |
| | Iphone 3 | 3.96 | 0 | 0 | **95.89** | 0.14 | 4.78 | 0 | 0 | **90.29** | 4.93 |
| | Nokia E71 | 0.92 | 0 | 0.34 | 0.76 | **97.98** | 0.02 | 0 | 0 | 1.34 | **98.64** |

### (b)  Markov approach

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **99.78** | 0.10 | 0.12 | 0 | 0 | **70.41** | 0 | 0.71 | 0.78 | 28.10 |
| | HTC HD2 | 0.53 | **99.36** | 0 | 0.11 | 0 | 6.51 | **60.11** | 0 | 2.40 | 30.98 |
| | Huawei U8150 | 0.77 | 0 | **99.16** | 0.05 | 0.02 | 15.50 | 0 | **46.14** | 1.36 | 37.00 |
| | Iphone 3 | 1.82 | 0.32 | 0.39 | **97.46** | 0 | 14.04 | 0.036 | 0 | **46.46** | 39.46 |
| | Nokia E71 | 0.22 | 0.04 | 0.04 | 0.22 | **99.48** | 3.96 | 0 | 0 | 0.82 | **95.22** |

Table III-5-2. Table Mean testing accuracy over 100 experiments based on type I and type II images

(a)  our approach

|  |  | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **98.88** | 0.12 | 0.73 | 0.01 | 0.26 | **97.01** | 0.03 | 2.28 | 0 | 0.68 |
|  | HTC HD2 | 0.04 | **99.32** | 0.17 | 0.18 | 0.29 | 0.78 | **94.53** | 0.42 | 1.95 | 2.31 |
|  | Huawei U8150 | 0.40 | 0.30 | **99.28** | 0 | 0.02 | 3.04 | 0.11 | **96.53** | 0.17 | 0.15 |
|  | Iphone 3 | 1.47 | 0.32 | 0 | **96.71** | 1.50 | 1.04 | 1.79 | 0.48 | **90.20** | 6.50 |
|  | Nokia E71 | 0.83 | 0.64 | 0.01 | 0.70 | **97.82** | 0.20 | 4.52 | 0 | 2.55 | **92.73** |

(b)  Markov approach

|  |  | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **94.76** | 0.66 | 4.16 | 0.34 | 0.08 | **76.17** | 0 | 3.09 | 0 | 20.74 |
|  | HTC HD2 | 0.93 | **94.16** | 1.40 | 1.52 | 1.99 | 1.40 | **65.56** | 0.54 | 0.40 | 32.10 |
|  | Huawei U8150 | 4.46 | 0.36 | **94.35** | 0.24 | 0.60 | 4.05 | 0.03 | **63.68** | 0 | 32.24 |
|  | Iphone 3 | 1.00 | 0.88 | 0.91 | **95.93** | 1.29 | 1.14 | 0.31 | 0.27 | **51.52** | 46.77 |
|  | Nokia E71 | 0.46 | 1.10 | 0.27 | 0.74 | **97.43** | 0.69 | 1.04 | 0 | 0.42 | **97.85** |

Table III-5-3. Mean testing accuracy over 100 experiments based on type III images with scale factor of 2

(a) our approach

|  |  | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **91.90** | 0.27 | 6.13 | 1.54 | 0.15 | **72.24** | 0 | 1.08 | 2.41 | 24.27 |
|  | HTC HD2 | 1.40 | **98.13** | 0.38 | 0 | 0.09 | 0.98 | **67.64** | 0.11 | 4.27 | 27.00 |
|  | Huawei U8150 | 13.05 | 0.02 | **86.14** | 0 | 0.79 | 15.61 | 0.07 | **55.04** | 3.30 | 25.98 |
|  | Iphone 3 | 3.71 | 0.64 | 0.14 | **95.07** | 0.43 | 0.61 | 0.79 | 0 | **76.93** | 21.68 |
|  | Nokia E71 | 1.92 | 0.14 | 1.22 | 0.36 | **96.36** | 0.62 | 0 | 0.08 | 1.86 | **97.44** |

(b) Markov approach

|  |  | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **99.25** | 0.02 | 0.69 | 0.03 | 0 | **99.70** | 0 | 0.30 | 0 | 0 |
|  | HTC HD2 | 0.04 | **99.58** | 0.33 | 0.02 | 0.02 | 73.38 | **26.49** | 0.13 | 0 | 0 |
|  | Huawei U8150 | 0.43 | 0.82 | **98.33** | 0.05 | 0.36 | 73.34 | 0 | **26.66** | 0 | 0 |
|  | Iphone 3 | 2.82 | 1.00 | 0.21 | **95.82** | 0.14 | 78.82 | 0 | 0 | **21.18** | 0 |
|  | Nokia E71 | 0.26 | 0.16 | 0.06 | 0 | **99.52** | 76.48 | 0 | 0 | 0 | **23.52** |

Table III-5-4. Mean testing accuracy over 100 experiments based on type IV images with scale factor of 0.5

(a)  our approach

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **82.43** | 1.20 | 6.66 | 6.88 | 2.83 | **77.15** | 1.05 | 14.83 | 2.01 | 4.95 |
| | HTC HD2 | 1.11 | **89.27** | 3.13 | 4.62 | 1.87 | 2.62 | **77.04** | 5.73 | 3.51 | 11.09 |
| | Huawei U8150 | 5.88 | 1.80 | **83.38** | 8.13 | 0.82 | 9.41 | 0.63 | **83.61** | 1.98 | 4.38 |
| | Iphone 3 | 18.32 | 12.00 | 14.25 | **42.71** | 12.71 | 19.29 | 11.96 | 12.18 | **28.96** | 27.61 |
| | Nokia E71 | 2.56 | 5.48 | 1.38 | 3.24 | **87.34** | 6.16 | 12.74 | 3.18 | 2.60 | **75.32** |

(b)  Markov approach

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **60.58** | 7.29 | 16.95 | 9.08 | 6.10 | **92.78** | 0 | 1.15 | 0 | 6.07 |
| | HTC HD2 | 4.27 | **67.53** | 5.22 | 10.24 | 12.73 | 90.87 | **0.96** | 1.02 | 0 | 7.16 |
| | Huawei U8150 | 16.36 | 10.11 | **60.20** | 7.93 | 5.41 | 92.16 | 0 | **2.12** | 0 | 5.71 |
| | Iphone 3 | 20.07 | 23.86 | 12.46 | **17.50** | 26.11 | 91.25 | 0 | 1.14 | **0** | 7.61 |
| | Nokia E71 | 7.30 | 24.86 | 5.48 | 10.02 | **52.34** | 86.58 | 0.04 | 1.00 | 0 | **12.38** |

Table III-5-5. Mean testing accuracy over 100 experiments based on type VI images with scale factor of 0.5

(a)  our approach

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **78.65** | 0.90 | 11.46 | 5.73 | 3.27 | **78.46** | 0.76 | 14.24 | 1.91 | 4.62 |
| | HTC HD2 | 0.75 | **81.84** | 1.89 | 5.44 | 10.07 | 1.04 | **80.38** | 1.93 | 3.62 | 13.02 |
| | Huawei U8150 | 10.98 | 0.73 | **77.46** | 8.50 | 2.32 | 11.29 | 0.38 | **83.73** | 1.59 | 3.02 |
| | Iphone 3 | 19.11 | 10.46 | 10.39 | **43.68** | 16.36 | 21.39 | 12.21 | 10.82 | **29.14** | 26.43 |
| | Nokia E71 | 2.72 | 11.80 | 3.08 | 6.18 | **76.22** | 5.58 | 12.64 | 1.44 | 2.36 | **77.98** |

(b)  Markov approach

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **58.19** | 6.34 | 20.66 | 9.25 | 5.56 | **88.66** | 0 | 0.34 | 0 | 11.00 |
| | HTC HD2 | 3.89 | **74.51** | 2.78 | 6.35 | 12.47 | 85.16 | **2.44** | 0.07 | 0 | 12.33 |
| | Huawei U8150 | 14.43 | 4.23 | **72.48** | 4.95 | 3.91 | 87.55 | 0 | **2.34** | 0 | 10.11 |
| | Iphone 3 | 22.61 | 20.39 | 14.71 | **18.21** | 24.07 | 86.43 | 0 | 0.07 | **0** | 13.50 |
| | Nokia E71 | 7.52 | 23.90 | 7.30 | 7.94 | **53.34** | 82.30 | 0.02 | 0 | 0 | **17.68** |

Table III-5-6. Mean testing accuracy over 100 experiments based on all type images with all scale factors

(a) our approach

|  |  | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **80.11** | 0.79 | 12.07 | 1.77 | 5.27 | **89.02** | 0.43 | 9.20 | 0.14 | 1.21 |
|  | HTC HD2 | 1.23 | **84.31** | 1.58 | 5.94 | 6.94 | 0.88 | **94.92** | 0.59 | 1.63 | 1.98 |
|  | Huawei U8150 | 10.47 | 0.60 | **82.67** | 3.35 | 2.92 | 5.20 | 0.36 | **93.35** | 0.30 | 0.79 |
|  | Iphone 3 | 8.22 | 5.75 | 6.40 | **70.15** | 9.48 | 4.86 | 2.14 | 4.91 | **81.30** | 6.79 |
|  | Nokia E71 | 5.50 | 8.43 | 1.85 | 8.40 | **75.83** | 2.01 | 2.97 | 1.17 | 1.91 | **91.95** |

(b) Markov approach

|  |  | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 | HTC G3 | HTC HD2 | Huawei U8150 | Iphone 3 | Nokia E71 |
| Actual brand | HTC G3 | **71.48** | 5.39 | 18.70 | 2.06 | 2.37 | **91.90** | 0.20 | 7.07 | 0.00 | 0.83 |
|  | HTC HD2 | 6.89 | **78.14** | 1.44 | 8.49 | 5.04 | 8.25 | **77.69** | 9.35 | 0.53 | 4.17 |
|  | Huawei U8150 | 15.43 | 1.81 | **78.47** | 2.64 | 1.65 | 15.04 | 0.23 | **83.99** | 0.05 | 0.69 |
|  | Iphone 3 | 7.17 | 15.25 | 6.46 | **62.28** | 8.83 | 19.42 | 2.02 | 20.15 | **51.01** | 7.40 |
|  | Nokia E71 | 4.31 | 9.12 | 5.03 | 5.16 | **76.38** | 9.83 | 1.70 | 9.94 | 1.38 | **77.16** |

*III-5-b. Smartphone source and post-capture identification with/without clustering*

## A. Statement of Results:

*A.1. Results without clustering*

The experimental results shown in III-5-a have demonstrated that the detection performance varies across different manipulations, in other words, some manipulation may lead us to the wrong judgment of the smartphone source if we are not aware of the manipulation. From the perspective of image forensics, it is very necessary and of great value to identify these different manipulations as well as to recognize the smartphone sources. Therefore, we mix the 65-class processed smartphone images together, and aim to simultaneously detect the smartphone source and the manipulation imposed on the processed images. In this detection, like the experiments in smartphone source identification, we randomly select 60% images from each class as training data and the remainders are used for testing in each of 100 total experiments. The mean values of the confusion matrix on the 65 classes over 100 experiments are shown by Figure III-1 (a) and Figure III-1(b) by using linear SVM and RBF-kernel SVM with our approach. Figure III-1(c) and (d) show the detection results by using Markov approach. The average hit

accuracy for the 65 classes is 42.1% with linear LibSVM and 43.5% with RBF LibSVM by using our approach, and 41.9% with linear LibSVM and 27.7% with RBF LibSVM by using 486-dimensional Markov approach.

As illustrated in Figure III-5-1, the performance of our approach is highly encouraging, compared to the average hit rate under random guess (i.e., 1/65 or 1.5%) for each class. Some hit rates are not so high, for instance, in the discrimination of class 49 and class 51, representing the images from iphone 3 in type V and type VI with the scale factor 0.5. In our opinion, some useful information to identify the smartphone source or keep track of a certain manipulation may be destroyed or covered by some other manipulations, such as the trace of cropping may be removed or covered by down-scaling and/or recompressed. In such case, the effect of cropping is trivial and could be ignored, resulting in the disability to distinguish between class 49 (iphone3 images in type V of scale parameter 0.5, with cropping) and class 51 (iphone3 images in type VI of scale parameter 0.5, without cropping). In this manner, it is not precise to label all type images into 65 classes, just based on the different combinations of the manipulations and smartphone sources. Therefore, it could be more reasonable for us to first employ clustering analysis to these 65 class smartphone image data, and then re-label these processed images based upon the clustering result.

*A.2. Results with clustering*

As we discussed, some post-capture operations, such as double compression followed by down-scaled will remove original traces/patterns of smartphone images, in such case, it does not make sense to classify these processed images after same/similar operations from different smartphones into different classes based on original smartphone types. Therefore, it is necessary for us to re-label and shrink original 65-class data set. First, we obtain the average values of the feature vectors from training data, and apply hierarchical agglomerative clustering (HAC) to the average feature vectors, originally labeled by the numbers from 1 to 65. Figure III-5-2 shows the hierarchical binary clustering tree by using single linkage algorithm to the pairwise distance measured by standardized Euclidean distance and usual Euclidean distance, respectively. As shown in the clustering tree (i.e., dendrogram), original class pairs, 30 and 32, 4 and 6, 43 and 45, 56 and 58, 17 and 19, are clustering together with the minimal distances, corresponding to the images in types III and IV, with the scale factor 2. Classes 8, 10, and 12 are clustering together, corresponding to HTC G3 images, manipulated by type IV, V and VI operations with the scale factor 0.5. It shows that interpolation (rescale) operation generates a major functionality across different types of manipulation. According to the two different pairwise distances, we re-label the original 65-class data into 18 classes, as shown in Figure III-5-2(a) and Figure III-5-2(b), respectively. For example, in Figure III-5-2(a), we integrate original classes, 30, 32, 31, 37, 38, 4, 6, 5, 11, and 13 together as a new class 1, original classes, 1, 2, 3, and 9 into new class 2, and so on. In Figure III-5-2(b), the original classes 30 and 32 are mixed together as new class 1, original classes 4 and 6 are integrated as new class 2, and so on. Table III-5-7 lists the clustering and original classes.

Figure III-5-3 and Figure III-5-4 illustrate the average values of confusion matrix over 100 experiments on the relabeled 18 class data. The experimental procedures are the same as those of previous experiments.  The comparison of the experimental results in Figure III-5-3 and Figure III-5-4 shows that the re-labeling from the clustering measured by Euclidean distance is better than the re-labeling with the standardized Euclidean distance.

In Figure III-5-3, most images in re-labeled class 5 are classified as re-labeled classes 7 and 3, and most images in re-labeled class 12 are detected as 13. In other words, the recognition of the images captured by Nokia E71 in type IV with the scale factor 0.3 (original class 59) were mostly recognized as the cluster consisting of original classes 8, 10, 12, and 46 (Iphone 3 in type IV with the scale factor 0.3) , and the cluster consisting of original classes 54, 55, 61, 60 (E71 in type IV with the scale factor 0.5) , 62 (E71 in type V with the scale factor 0.5), and 64 (E71 in type VI with the scale factor 0.5). Similarly, it is not easy to accurately identify the images captured by Huawei U8150 in type IV with the scale factor 0.3 (original class 33).

In Figure III-5-4, the re-labeled class 11 is prone to be classified as re-labeled classes 7 and 10.  That is, the images captured by Huawei U8150 in type IV with the scale factor 0.3 (original class 33) may not be accurately identified.

To compare the detection results shown by Table III-5-6, wherein image labeling is based on the five smartphone sources but ignored the manipulations, we re-label all types of images with all scale factors into five clusters (classes), the cluster 1 consists of original classes 30, 32, 4, 6, …, 43,  45, 56 and 58;  cluster 2 consists of original classes 24 and 26; cluster 3 consists of original classes 17 and 19; cluster 4 only contains original class 14; and cluster 5 is derived from original class 53. Table III-5-9 shows the mean testing accuracy of confusion matrix over 100 experiments. As before, 60% images are randomly selected from each cluster for training and the remaining are used for testing. By comparing the experimental results shown in Table III-5-7 and in Table III-5-8, the advantage is noticeable by considering smartphone source and manipulation together with the aid of clustering analysis (i.e., followed by supervised learning).

## B.    Tables:

Table III-5-7 Clustering by standardized Euclidean distance and original classes

| Cluster | Original classes in the cluster | Cluster | Original classes in the cluster | Cluster | Original classes in the cluster |
|---|---|---|---|---|---|
| 1 | 4, 5, 6, 11, 13, 30, 31, 32, 37, 39 | 7 | 54, 55, 60, 61, 62, 64 | 13 | 34, 36, 38 |
| 2 | 1, 2, 3, 9 | 8 | 57, 63, 65 | 14 | 24, 26 |
| 3 | 8, 10, 12, 46 | 9 | 40, 41, 42, 43, 44, 45, 47, 48, 49, 50, 51, 52 | 15 | 56, 58 |
| 4 | 7 | 10 | 27, 28, 29, 35 | 16 | 17, 19 |
| 5 | 59 | 11 | 18 | 17 | 14 |
| 6 | 15, 16, 20, 21, 22, 23, 25 | 12 | 33 | 18 | 53 |

Table III-5-8 Clustering by Euclidean distance and original classes

| Cluster | Original classes in the cluster | Cluster | Original classes in the cluster | Cluster | Original classes in the cluster |
|---|---|---|---|---|---|
| 1 | 30,32 | 7 | 34,36,38 | 13 | 43,45 |
| 2 | 4,6 | 8 | 57,65 | 14 | 56,58 |
| 3 | 5,11,13,31,37,39 | 9 | 50,52 | 15 | 24,26 |
| 4 | 27 | 10 | 7,15,16,20,21,22,23,25,40,41,42, 44,47,48,49,51,54,55,59,60,62,64 | 16 | 17,19 |
| 5 | 1,2,3,8,9,10,12 | 11 | 33 | 17 | 14 |
| 6 | 28,29,35 | 12 | 18,63 | 18 | 53 |

Table III-5-9. Mean testing accuracy over 100 experiments based on all type images with all scale factors

| | | Prediction outcome (%) using linear kernel | | | | | Prediction outcome (%) using RBF kernel | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 |
| | Cluster 1 | **99.90** | 0.08 | 0.01 | 0 | 0.00 | **99.81** | 0.19 | 0 | 0 | 0.00 |
| | Cluster 2 | 10.32 | **89.68** | 0 | 0 | 0 | 13.65 | **86.35** | 0 | 0 | 0 |
| Actual cluster | Cluster 3 | 1.20 | 0 | **98.80** | 0 | 0 | 0.52 | 0 | **99.48** | 0 | 0 |
| | Cluster 4 | 0.09 | 0 | 0 | **99.91** | 0 | 0.44 | 0 | 0 | **99.56** | 0 |
| | Cluster 5 | 4.12 | 0.24 | 0.42 | 0 | **95.22** | 6.16 | 0 | 0 | 0 | **93.84** |

## C.    Figures:



(a)

(b)

(c)

(d)
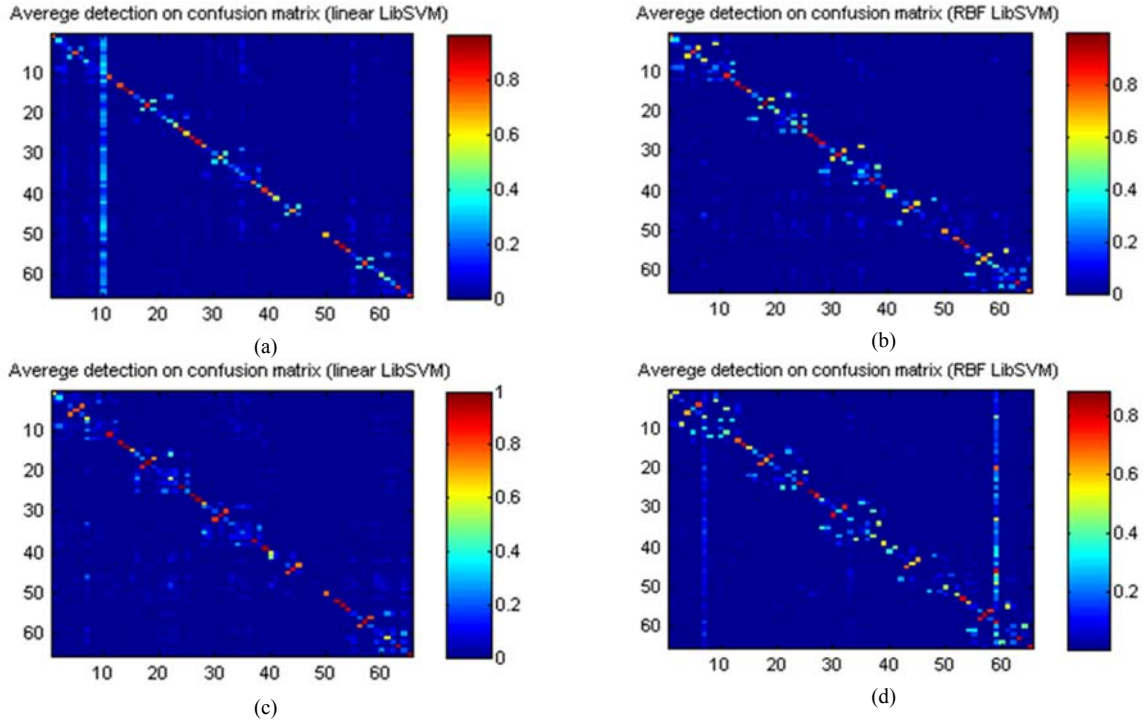
**Figure III-5-1**. Mean confusion matrix over 100 experiments using our approach (a and b) and using Markov approach (c and d).

The hierarchical binary cluster tree

(pairwise distance measured by standardized Euclidean distance)

(a)



The hierarchical binary cluster tree
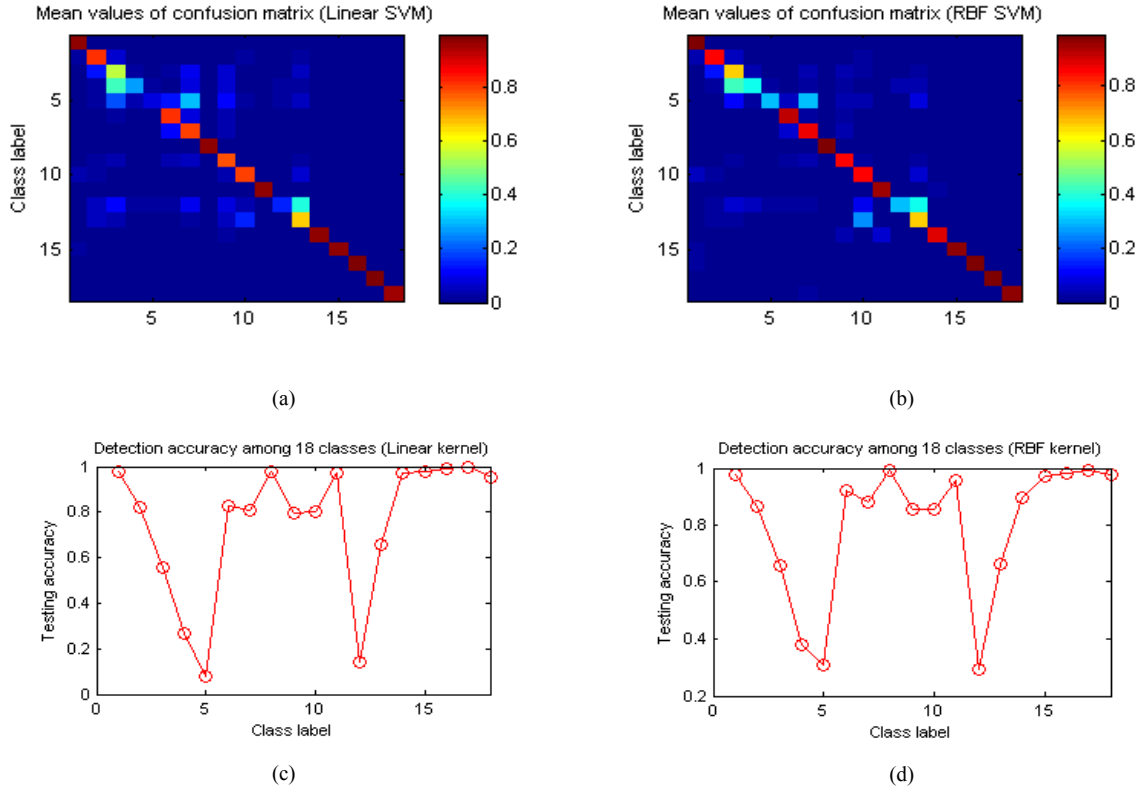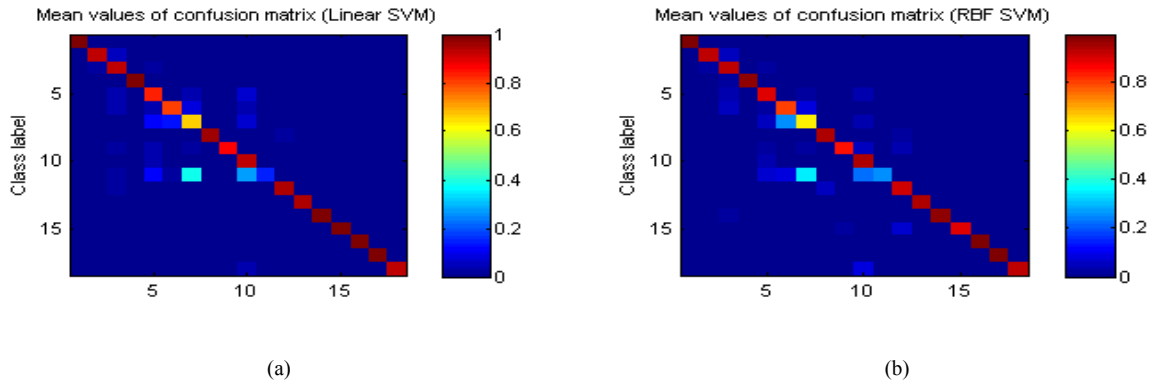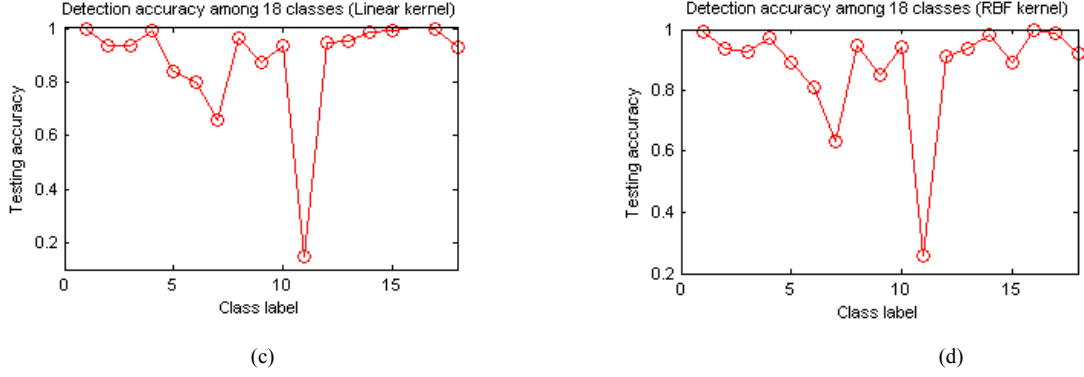
(pairwise distance measured by Euclidean distance)
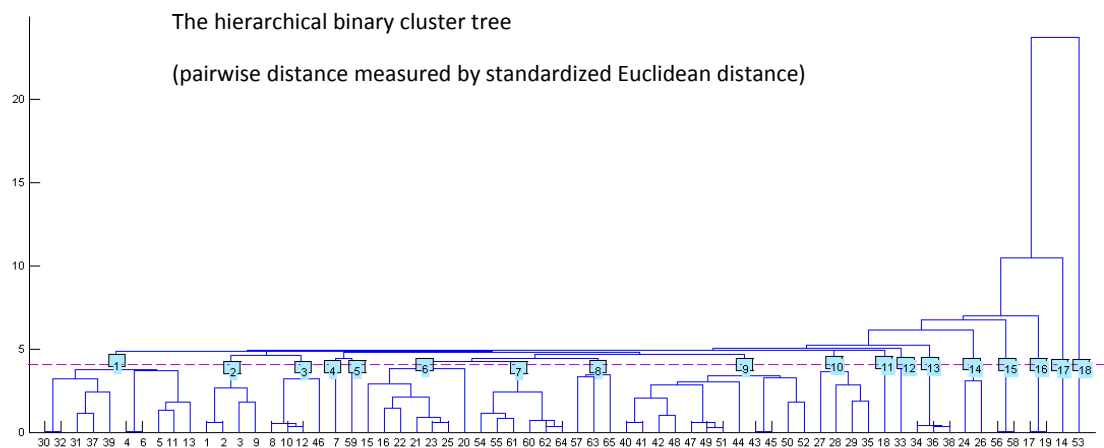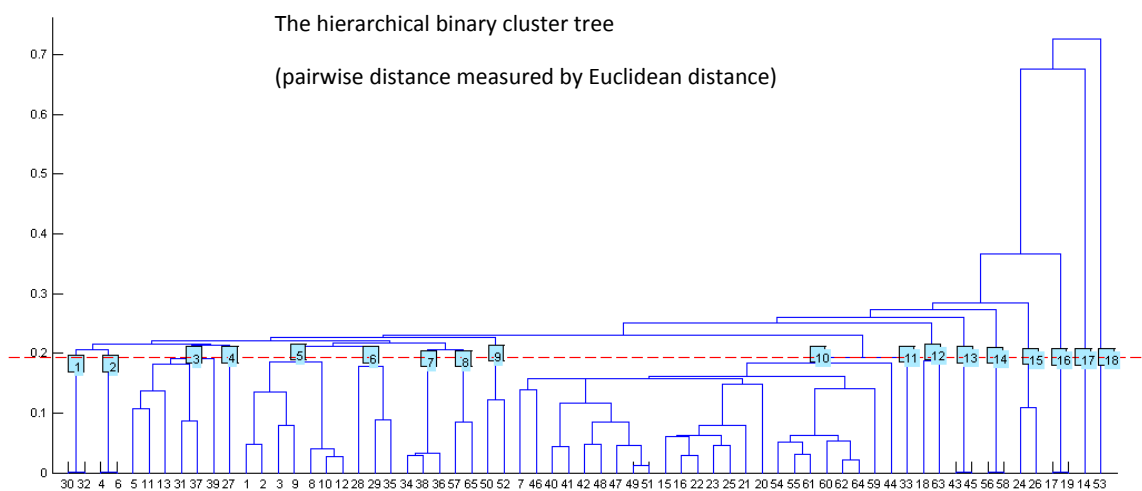
(b)

Figure III-5-2. Hierarchical binary cluster tree measured by standardized Euclidean distance (a) and Euclidean distance (b). Note that the numbers in the squares represent re-labeled classes.

(a)                            (b)

(c)                            (d)

**Figure III-5-3**. Mean confusion matrix over 100 experiments ((a) and (b)) and the hit rate for re-labelled 18 classes ((c) and (d)), based on HAC with pairwise distance measured by standardized Euclidean distance in Figure III-2(a)

(a)                            (b)

**Figure III-5-4**. Mean confusion matrix over 100 experiments ((a) and (b)) and the hit rate for re-labelled 18 classes ((c) and (d)), based on HAC with pairwise distance measured by Euclidean distance in Figure III-2(b)

## III-6. Detection of MPEG Double Compression

### A.     Statement of Results:

*A.1. Detection of MPEG double compression with same encoder*

Table III-6-1 and Table III-6-2 show the experimental results when one MPEG-2 encoder is utilized to simulate the double compression process at different target output bit-rates. It turns out that the detection accuracy of the detector can be achieved approximately above 87.00%.  In these two MPEG-2 encoders, the CBR scheme in the TM5 is more flexible than that in the Premiere. Even when the target output bit-rate of the secondary compression is fewer than that of the primary compression, TM5 will still select small quantization scale factors to quantize a number of DCT coefficients, which can be used to realize the double compression detection. Given a video clip compressed by TM5, the detectors are easier to identify whether it is doubly compressed. Since Wang's method only works for double MPEG-2 compression with VBR mode, we will only compare our method with Chen's scheme to detection of double MPEG-2 compressed videos for CBR mode. Both two schemes can have excellent detection results when the target output bit-rate of the secondary compression is very high. However, when the target output bit-rate of the secondary compression is lower than that of the primary compression, the detection accuracy of both of them will decrease, but the simulation results also show that the robustness of our scheme is considerably better than Chen's scheme.

*A.2. Detection of MPEG double compression with different video encoders*

In this experiment, two DVs (digital video camcorders), Sony HDR-XR500E and Canon FS10E, are utilized to obtain the original MPEG-2 videos. We use each camcorder to record 50 nature video clips with length of 300 frames in our campus. In each test group, 50 nature image sequences are firstly encoded into MPEG-2 compressed files by the built-in encoder in the DV. These original compressed streams are input into PC, decoded and re-coded into 50 double compressed streams by the MPEG-2 video coding software -- Adobe Premiere Pro 2.0 or TM5.  Finally 200 test streams (100 single compressed streams and 100 double compressed streams) will be put into

our detector to test its performance.

In the parameter settings, the most important parameters in DVs are resolution and target output bit-rate. The standard definition video format is selected as our encoding mode, whose resolution is 720×576 and output bit-rate is 6 Mbps. The other parameters just only affect the subjective effects of video resources, but have less impact on the statistical characteristics of DCT coefficient distribution, and we initial them as default values in the DVs. In the MPEG-2 software coders, all parameters are set the same as the MPEG-2 encoders in Section V.A. We only adjust the target output bit-rate of software encoders to create new video files with different quality to test the adaptability of our detection scheme.

The detection accuracy and recall for doubly MPEG-2 compressed videos are shown in Table III-6-3 and Table III-6-4, where the target output bit rate of the secondary compression varies from 4 Mbps to 8 Mbps in steps of 1 Mbps. The detection results in these two tables show that our proposed method indeed outperforms Chen's method when the primary compression and secondary compression adopt different MPEG-2 encoders, because the quantization process, the motion prediction algorithm, and the bit-rate control scheme are almost different between these two compression processes. Those differences may induce the rearrangement of DCT coefficients in the doubly compressed video and the first digit distribution of AC coefficients may also tend to obey the Benford's law. It is also found that the detection performance improves as the output bit-rate of the secondary compression increases. When the target output bit-rate of the secondary compression is lower than that of the primary compression, implied that the number of macroblocks quantized by small quantization scale factors is reduced, the quantization process with large quantization scale factors may weaken the features of the primary compression. Thus, the performance of double compression detector declines. On the other hand, some test sequences with simple contents and slow movements often incur detection errors, because most of original DCT coefficients in those frames focus on a certain interval, and their distribution is not consistent with the Laplacian distribution assumption.

*A.3. Detection of double compression with frame tampering operation*

Frame tampering is one of the common video forgery operations, which can change the video content and confuse the viewers by removing some special frames in the video resources, such as some surveillance videos. In this experiment, 50 original compressed videos recorded by Sony HDR-XR500E at 6 Mbps are decoded into an image sequences, and the first three images are removed to simulate the frame tampering operation. Finally these doctored sequences are re-coded at different target output bit-rates by TM5, and 100 test streams (50 singly compressed streams and 50 doubly compressed streams) will be put into our detector to verify its robustness.

Table III-6-5 shows the experimental results that the performance of our detector has declined slightly. When some frames are removed, the structure of GOP in the original sequence will be damaged, and the type of some subsequent frames will be changed in the secondary compression, i.e. some inter frames in the primary compression process may be re-coded as intra frames during the secondary compression. Since the distribution of DCT coefficients in inter-frames behaves similarly as that in intra-frames, there still are some convex patterns in the distribution of quantized DCT coefficients.

## B.    Tables:

Table III-6-1.  The Detection Performance of Double Compression System with TM5

| First Encoder (bit-rate) | Second Encoder (bit-rate) | Chen's scheme | | Proposed scheme | |
|---|---|---|---|---|---|
| | | Precision | Recall | Precision | Recall |
| TM5(6 Mbps) | TM5(4 Mbps) | 62.50% | 60.00% | 89.36% | 84.00% |
| | TM5(5 Mbps) | 69.23% | 72.00% | 93.33% | 84.00% |
| | TM5(7 Mbps) | 90.57% | 96.00% | 100.00% | 100.00% |
| | TM5(8 Mbps) | 100.00% | 100.00% | 100.00% | 100.00% |

Table III-6-2.  The Detection Performance of Double Compression System with Premiere

| First Encoder (bit-rate) | Second Encoder (bit-rate) | Chen's scheme | | Proposed scheme | |
|---|---|---|---|---|---|
| | | Precision | Recall | Precision | Recall |
| Premiere(6 Mbps) | Premiere (4 Mbps) | 64.58% | 62.00% | 86.96% | 80.00% |
| | Premiere (5 Mbps) | 68.75% | 66.00% | 87.50% | 84.00% |
| | Premiere (7 Mbps) | 88.24% | 90.00% | 100.00% | 96.00% |
| | Premiere (8 Mbps) | 100.00% | 100.00% | 100.00% | 96.00% |

Table III-6-3.  The Detection Performance of Double Compression System with Different Encoders

| First Encoder | Second Encoder | Chen's scheme | | Proposed scheme | |
|---|---|---|---|---|---|
| | | Precision | Recall | Precision | Recall |
| Sony HDR-XR500E (6 Mbps) | TM5(4 Mbps) | 52.38% | 44.00% | 95.80% | 92.00% |
| | TM5(5 Mbps) | 57.14% | 56.00% | 100.00% | 96.00% |
| | TM5(6 Mbps) | 61.70% | 58.00% | 100.00% | 96.00% |
| | TM5(7 Mbps) | 62.50% | 60.00% | 100.00% | 96.00% |
| | TM5(8 Mbps) | 64.71% | 66.00% | 100.00% | 100.00% |
| Canon FS10E (6 Mbps) | TM5(4 Mbps) | 67.31% | 70.00% | 88.50% | 92.00% |
| | TM5(5 Mbps) | 66.67% | 68.00% | 92.00% | 92.00% |
| | TM5(6 Mbps) | 66.67% | 72.00% | 95.80% | 92.00% |
| | TM5(7 Mbps) | 68.52% | 74.00% | 96.00% | 96.00% |
| | TM5(8 Mbps) | 69.39% | 68.00% | 96.00% | 96.00% |

Table III-6-4.  The Detection Performance of Double Compression System with Different Encoders

| First Encoder | Second Encoder | Chen's scheme | | Proposed scheme | |
|---|---|---|---|---|---|
| | | Precision | Recall | Precision | Recall |
| Sony HDR-XR500E (6 Mbps) | Premiere (4 Mbps) | 57.69% | 60.00% | 100.00% | 92.00% |
| | Premiere (5 Mbps) | 56.52% | 52. 00% | 100.00% | 96.00% |
| | Premiere (6 Mbps) | 62.96% | 68.00% | 100.00% | 100.00% |
| | Premiere (7 Mbps) | 63.64% | 70.00% | 100.00% | 100.00% |
| | Premiere (8 Mbps) | 64.91% | 74.00% | 100.00% | 100.00% |

| | | | | | |
|---|---|---|---|---|---|
| Canon FS10E | Premiere (4 Mbps) | 63.83% | 60.00% | 95.70% | 88.00% |
| | Premiere (5 Mbps) | 69.30% | 68.00% | 95.80% | 96.00% |
| | Premiere (6 Mbps) | 70.59% | 72.00% | 96.00% | 96.00% |
| (6 Mbps) | Premiere (7 Mbps) | 70.59% | 72.00% | 100.00% | 96.00% |
| | Premiere (8 Mbps) | 71.15% | 74.00% | 100.00% | 96.00% |

Table III-6-5. The Detection Performance of Frame Tampering Operation

| First Encoder | Second Encoder | Precision | Recall |
|---|---|---|---|
| | TM5 (4Mbps) | 71.88% | 92.00% |
| Sony HDR-XR500E (6 Mbps) | TM5 (5 Mbps) | 88.00% | 88.00% |
| | TM5 (6 Mbps) | 87.50% | 84.00% |
| | TM5 (7 Mbps) | 96.00% | 96.00% |
| | TM5 (8 Mbps) | 100.00% | 84.00% |

## III-7. MP3 steganalysis

## A. Statement of Results:

### A.1. Statistics of Feature Sets

We compared the significance of GGD shape statistical features, frequency-based subband moment statistical features, accumulative Markov transition features and neighboring joint density features, respectively (Qiao, Sung and Liu 2013).

Figures III-7-1(a) and (b) list the F scores of the ANOVA of the features extracted from covers and steganograms produced by using the MP3Stego audio steganographic tool with 16% and 20% modification density. The Y-axis indicates the F score, and the X-axis gives the number of features.

From the comparison of the F scores, we found that frequency-based subband moment statistical features outperform the other feature sets, especially those extracted from middle frequency and correspond to higher F scores. We surmised that the detection performance using frequency-based subband moment statistical features is the best. Accumulative Markov transition features and neighboring joint density features obtain similar F scores at 16% and 20% modification density. In GGD shape statistical features, the high-order moment statistics, especially the skewness of shape parameters, are more discriminative. Although the higher F score indicates the more significant feature, the interaction and the redundancy of the feature sets also affect the classification performance. Therefore, the testing accuracy is more reliable in evaluating the performance of features.

Figures III-7-2(a) and (b) illustrate the comparison of SVM testing accuracies by using all samples at 16% and 20% modification density correspondingly. The detection performance of frequency-based subband moment statistical features is superior to the performance of accumulative neighboring joint density features, GGD shape statistical features, and accumulative Markov transition features. Moreover, the distribution of the testing accuracy of frequency-based subband moment statistical features shows a smaller degree of dispersion than other feature sets, which indicates a more stable classification.

*A.2. Comparison of Feature Selection Methods*

We combined GGD shape statistical features, frequency-based subband moment statistical features, as well as accumulative Markov transition features and accumulative neighboring joint density features to form a comprehensive approach of MP3 audio steganalysis. However, the large feature dimension and redundancy compromise the performance and the efficiency of the detection. To further increase the classification accuracy, we employed two widely used feature selection methods; ANOVA and SVM-RFE. We also designed a two-step approach by combining these two methods. For the two-step approach, we picked 200 as a threshold to divide the processes of two feature selection methods. All features were ranked using ANOVA, The features with the top 200 F scores were chosen as the input for SVM-RFE. The size of feature subset for classification continued increasing from 1 to 200 in the order of the feature rank provided by SVM-RFE. For accurate evaluation, we divided the whole data set into low ($\beta < 0.162$), middle ($0.151 <= \beta <= 0.171$), and high ($\beta >= 0.162$) complexity zones with 50% overlap between adjacent zones using GGD shape parameter of all quantized MDCT coefficients in each sample. Furthermore, the whole dataset, including samples with all complexities, was used as another category.

Regarding the relation between detection performance and signal complexity, as shown in Table 1, as the signal complexity increases, the detection performance decreases. Since the average signal complexity of the whole dataset is 0.164, the average classification accuracy of all samples is close to the accuracy of middle complexity.

In the comparison of the three feature selection methods, the two-step approach outperforms ANOVA and SVM-RFE in each category of signal complexity and modification density. Our study also shows that the two-step approach adopts the advantage of low standard errors and thus provides more stable detection performance.

In addition to the comparison shown in Table III-7, the receiver operating characteristic (ROC) curves by using ANOVA, SVM-RFE, and the two-step approach are given in Figure III-7-3. The ROC curve of the two-step approach generates the largest area under the curve at 20% modification density.

## B. Tables:

Table III-7

Average Testing Accuracy Values and Standard Errors of Feature Selection Methods: ANOVA, SVM-RFE, and Two-step Approach Incorporating ANOVA with SVM-RFE at the Optimal Feature Dimension.

| Modification density | Signal complexity | Testing accuracy (mean ± std, %) | | |
|---|---|---|---|---|
| | | ANOVA | SVM-RFE | ANOVA & SVM-RFE |
| 8% | Low | 81.6 ± 1.4 | 85.9 ± 2.8 | 88.6 ± 1.4 |
| | Middle | 78.1 ± 1.2 | 82.2 ± 2.3 | 84.9 ± 0.9 |
| | High | 75.6 ± 1.8 | 78.2 ± 1.9 | 81.4 ± 1.2 |
| | All | 79.4 ± 1.1 | 86.6 ± 1.6 | 86.1 ± 1.7 |
| 12% | Low | 88.6 ± 1.4 | 94.0 ± 2.2 | 95.1 ± 1.6 |
| | Middle | 84.9 ± 1.7 | 91.0 ± 1.8 | 91.3 ± 1.4 |
| | High | 82.6 ± 0.6 | 88.9 ± 2.0 | 90.0 ± 1.8 |
| | All | 85.0 ± 0.9 | 90.6 ± 1.2 | 90.4 ± 0.9 |
| 16% | Low | 90.4 ± 1.4 | 95.9 ± 2.8 | 97.0 ± 1.6 |
| | Middle | 89.3 ± 1.3 | 93.2 ± 2.1 | 94.8 ± 1.1 |
| | High | 87.7 ± 1.0 | 91.8 ± 2.2 | 92.1 ± 1.2 |
| | All | 88.5 ± 0.6 | 93.2 ± 0.7 | 93.3 ± 1.0 |
| 20% | Low | 94.6 ± 0.8 | 96.7 ± 2.6 | 98.6 ± 1.0 |
| | Middle | 91.1 ± 1.6 | 94.7 ± 2.4 | 95.3 ± 1.4 |
| | High | 89.9 ± 0.7 | 94.3 ± 2.8 | 93.6 ± 1.6 |
| | All | 91.0 ± 0.9 | 94.9 ± 0.5 | 95.6 ± 0.6 |

## C. Figures:
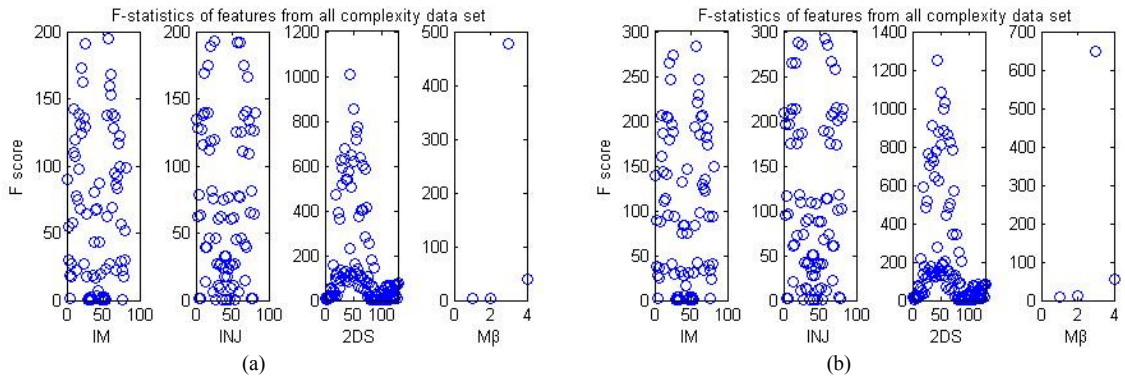


Figure III-7-1.  One way ANOVA F scores (Y-label) of Accumulative Markov transition features (IM), Accumulative neighboring joint density (INJ), frequency-based subband moment statistical features (2DS), and GGD shape statistical features (Mβ) from whole data set including samples with all signal complexities at 16% and 20% modification density in (a) and (b) respectively.
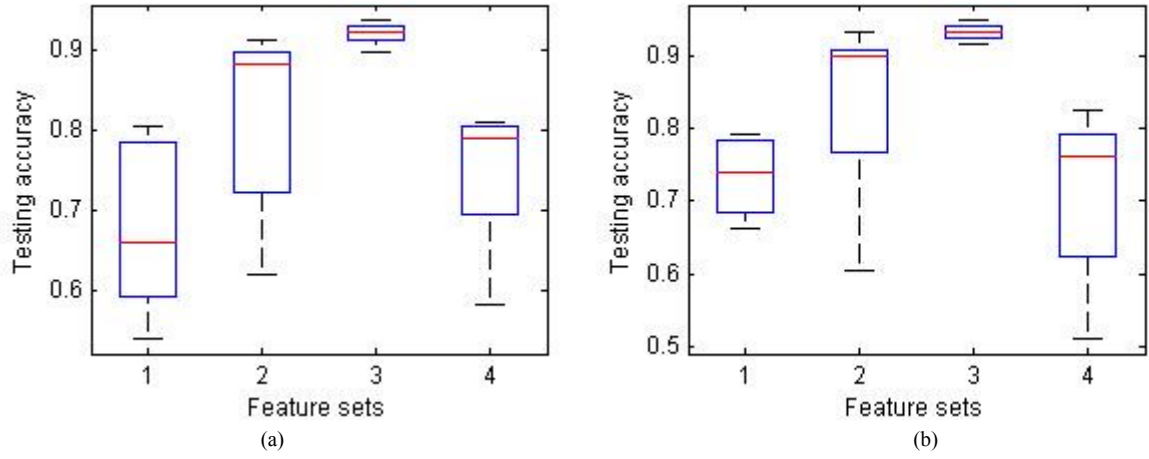
Figure III-7-2. SVM testing accuracies by using feature sets: Accumulative Markov transition features (1), Accumulative neighboring joint density features (2), Frequency-based subband moment statistical features (3), and GGD shape statistical features (4) from whole data set including samples with all signal complexities at 16% and 20% modification density in (a) and (b) respectively.
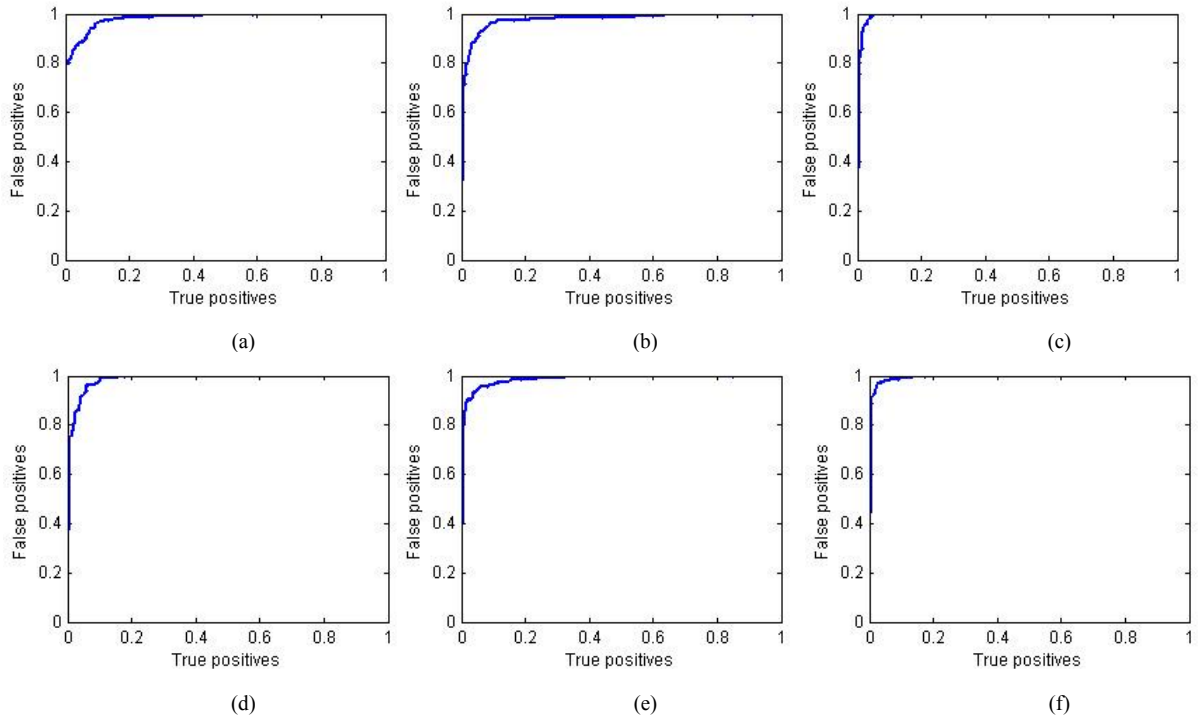


Figure III-7-3. ROC curves by using ANOVA (a,d), SVM-RFE (b,e), and two-step feature selection (c,f) methods, in detection of MP3 steganograms with 16% (first row) and 20% (second row) modification densities.

## III-8. AAC Audio Forgery Detection

### A. Statement of Results:

Figure III-8-1 shows the SRSC features extracted from an untouched AAC audio file and from the cropping by 50 samples and the cropping by 900 samples individually and recompressed versions.

We apply a popular SVM technique LibSVM (Chang and Lin 2011) with a linear kernel for training and testing. One hundred experiments are performed for training and testing. In each experiment, 60% feature sets from each category are randomly selected for training and the remainders are used for testing. The mean testing results over 100 experiments are listed in the confusion matrix, shown by Table III-8-1.

Table III-8-2 shows the confusion matrix with the experimental results over 100 experiments.

Figure III-8-2 shows the SRSC features extracted from the six parts of an original AAC audio file (first row) and from the six part of the doctored AAC audio file with the forgery taking place on the middle. The comparison show that the first three parts of the original audio and doctored audio are similar, but the pattern of the SRSC features from the last three parts are different, which approximately reveals the forged area in doctored AAC audio stream in the middle.

### B. Tables:

Table III-8-1. Confusion matrix on testing sets (mean values, %) by using LibSVM with linear kernel over 100 experiments.

| Prediction / Truth | untouched | Manipulation (cropped by) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 50 | 200 | 400 | 480 | 512 | 750 | 900 | 1000 |
| untouched | **99.4** | 0.2 | 0.0 | 0.0 | 0.0 | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 |
| Manipulation (Cropped by) — 5 | 3.4 | **96.6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 50 | 1.8 | 0 | **98.2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 200 | 2.1 | 0 | 0 | **97.9** | 0 | 0 | 0 | 0 | 0 | 0 |
| 400 | 1.5 | 0 | 0 | 0 | **98.6** | 0 | 0 | 0 | 0 | 0 |
| 480 | 1.2 | 0 | 0 | 0 | 0 | **98.8** | 0 | 0 | 0 | 0 |
| 512 | 2.3 | 0 | 0 | 0 | 0 | 0 | **97.7** | 0 | 0 | 0 |
| 750 | 2.0 | 0 | 0 | 0 | 0 | 0 | 0 | **98.1** | 0 | 0 |
| 900 | 2.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **97.7** | 0 |
| 1000 | 2.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **97.9** |

Table III-8-2. Confusion matrix on testing sets (mean values, %) by using LibSVM with linear kernel over 100 experiments.

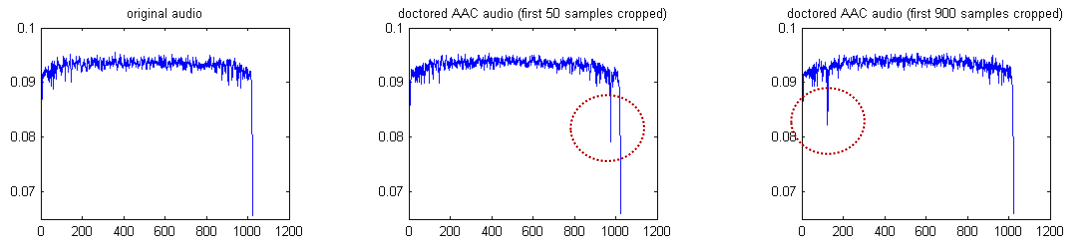| Prediction / Truth | untouched | forgery (shifted by) | | | | |
|---|---|---|---|---|---|---|
| | | 100 | 300 | 500 | 700 | 900 |
| untouched | **98.8** | 0.2 | 0.1 | 0.2 | 0.3 | 0.3 |
| forgery (shifted by) — 100 | 1.2 | **98.7** | 0 | 0 | 0 | 0.0 |
| 300 | 0.8 | 0.1 | **99.1** | 0 | 0 | 0.0 |
| 500 | 0.6 | 0 | 0.0 | **99.4** | 0.0 | 0.0 |
| 700 | 0.8 | 0 | 0.0 | 0 | **99.2** | 0.0 |
| 900 | 1.2 | 0.2 | 0.0 | 0.0 | 0 | **98.6** |

## C.  Figures:



Figure III-8-1. SRSC features of original AAC audio (a) and the AAC audio once cropped by 50 samples (b) and by 900 samples (c)
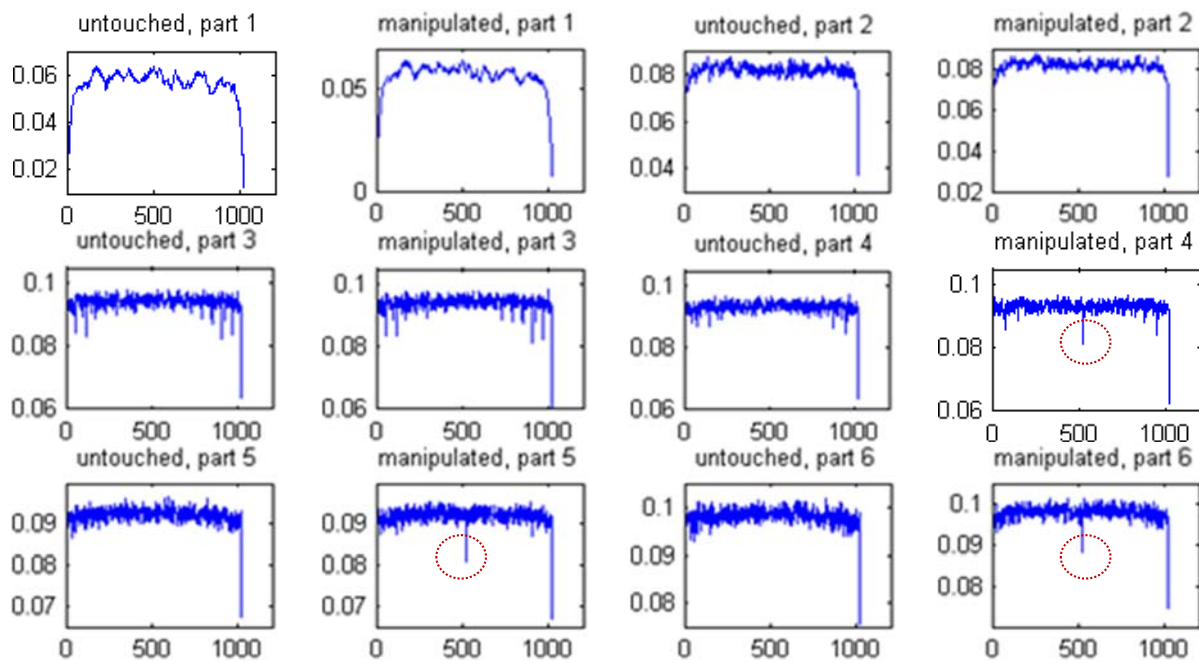


Figure III-8-2. The comparison of the SRSC features extracted from the six parts of an original AAC audio file and from manipulated AAC audio file with doctoring on the middle.

# IV.   Conclusions

## A.   Discussion of findings

### A.1. Steganalysis and forgery detection in JPEG images

In JPEG image steganalysis, compared to the calibration that only takes once-cropping (e.g., only shifting by 4 rows and 4 columns), the computation cost of our proposed 63-time-cropping-based calibration is relatively high but obtains a better detection accuracy.

It is worth noting that the 63-time-cropping-based approach is useful to generate the reference features for steganalysis and is also very promising to detect the misaligned cropping and recompression with the same quantization matrix and relevant forgery including copy-paste and composite forgery that are derived from the same camera source and encoded with the same quantization table (Liu 2011b).

In steganalysis of YASS, although Li, Shi and Huang (2009) demonstrated the weakness of the YASS steganographic system, the detection algorithm does not search all candidate host blocks, resulting in deteriorated detection performance when detecting the steganograms produced by a large B-block parameter. Additionally, the detection assumes the condition of the exact positions of AC coefficients that are used for data embedding, which is generally inapplicable. Following the strategy to expose potential candidate blocks, our study has surmounted such obstacles by searching all possible candidate blocks and comparing the neighboring joint density of these candidate blocks and the non-candidate neighboring blocks.

In an original YASS embedding algorithm, the embedding is limited to the 19 low-frequency AC coefficient; the upper-left of the first B-block is overlapped with the upper-left of the first 8×8 block. If we assume that the embedding positions of binary hidden bits are not limited to the 19 low-frequency AC DCT coefficients, our approach is still effective for the detection because our feature extraction is not limited to the position of 19 low-frequency AC coefficients. However, if prior knowledge of approximate embedding position is available, the detection performance may be further improved.

If the first B-block is randomly misplaced from the upper-left point of the first 8×8 block, we can exhaust all possibility of mismatching, a total of 64 combinations including the original exact matching; accordingly we can retrieve the *diff-absNJ* features in each mismatching which will detect such polymorphism of the YASS steganographic system. In this case, the detector will contain 64×63 = 4032 features. However, the detector cannot deal with the completely randomized embedding if we further revise and improve the YASS algorithm.

A rich model-based detector can be applied to detect YASS steganograms without exposing the position of candidate blocks, although the detection performance is not accurate as our approach, and the computational cost is also fairly high with an ensemble classifier and too high to be suitable with SVM. However, a rich-model-based approach

demonstrates a direction to deal with completely randomized embedding, which may be further investigated. Meanwhile, YASS detection is still difficult when the *noused* parameter is small.

To reduce the feature dimensionality and to further improve the detection accuracy, we may integrate all detectors together; a feature selection algorithm is applied to select optimal feature set. The feature selection to reduce feature dimensionality and improve detection accuracy in steganalysis has been studied in our previous research (Liu, Sung, Chen et al 2008; Liu et al 2010). We do not apply any feature selection algorithm in this study to compare the detection performance under different combinations of features. There are many algorithms to select optimal feature set and achieve the best classification performance, such as SVM-RFE (Guyon et al 2002); MSVM-RFE (Zhou and Tuck 2007), recursive feature addition based on supervised learning and similarity measurement (Liu, Sung, Chen et al 2010), minimum Redundancy Maximum Relevance (Peng et al 2005), etc. The steganalysis performance could be further improved by employing feature selection algorithms while obtaining an optimal feature set with reduced feature dimensionality, which could be applied to rich models.

In summary, our study shows that our approach has multiple promising applications in image forensics. Compared to the state-of-the-art of steganalysis detectors, our approaches deliver better or comparable detection performances with a much smaller feature set to detect several steganographic systems including DCT-embedding-based adaptive steganography and YASS. Our method is also effective to detect seam-carved forgery in JPEG images. By integrating calibrated neighboring density with spatial domain rich models that were originally designed for steganalysis, the hybrid approach obtains the best detection accuracy to discriminate seam-carved forgery from an untouched image in JPEG format. Our study shows that it is a promising manner by exploring steganalysis and forgery detection together.

*A.2. Detection of JPEG double compression and smartphone image source and post-capture processing*

While detecting JPEG double compression, experimental results show that the detection performance varies with different image quality. We analyze the impact of compression quality factor on the detection as follows. Let $c$ denote the DCT coefficient before quantization, $d_1$ denote the quantized DCT coefficient at the first JPEG compression with the quantization factor $q_1$, and $d_2$ denote the quantized DCT coefficient after the second compression with the quantization factor $q_2$, R(•) is a round function, e.g. R(3.5) = 4,

$$d_1 = R(c / q_1) \qquad\qquad\qquad (\text{IV-1})$$

$$d_2 = R(d_1 \, q_1 / q_2) \qquad\qquad\qquad (\text{IV-2})$$

The corresponding DCT coefficient $s$ in the single JPEG compression using the division factor $q_2$ is obtained by

$$s = R(c / q_2) \qquad\qquad\qquad (\text{IV-3})$$

Suppose that $c > 0$, the range of $d_1$ $q_1$ is $(c - 0.5q_1, c + 0.5q_1]$, and the range of $d_2 - s$ is

$$[R(c/q_2 - q_1/q_2) - R(c/q_2), R(c/q2 + q1/q2) - R(c/q2)]$$

(i) If the quality of the first compression equals the quality of the second compression, that is $q_1 = q_2$, then $d_2 = s$ ;

(ii) If the quality of the first compression is not equal to that of the second compression, the error range

$$d2 - s \in [-R(q_1/2q_2) - 1, -(q_1/2q_2) + 1] \tag{IV-4}$$

If the quality of the first compression is lower than that of the second compression, $q_1 > q_2$, then the scope of the error range spread out, and the error tends to become large; if the quality of the first compression is higher than that of the second compression, $q_1 < q_2$, then the scope of the error range is narrow. As a result, the detection of the double compression at $q_1 > q_2$ should be better than the detection at $q_1 < q_2$, which has been validated by our experimental results.

The relative error between $d_2$ and $s$ is listed as follows:

$$|(d_2 - s) / s| \leq \frac{R(q_1/q_2) + 1}{|R(c/q_2)|} \tag{IV-5}$$

In terms of image complexity, in general, high image complexity corresponds to the large shape parameter of the GGD of the DCT coefficients and the high probability of large DCT coefficient, that is, the value of $c$ tends to be large, resulting in large $s$, and the relative error tends to be small, therefore, the detection accuracy in high image complexity deteriorates.

While identifying smartphone image source and manipulation together, from our standpoint, some post-capture operations such as double compression followed by down-scaling will remove original traces/patterns of smartphone images, in such case, it is not good to classify these processed images into different classes based on original smartphone types. However, we did not know the particular operations that remove original traces, therefore, clustering is utilized to classify the smartphone images from different smartphones but processed by similar operations into the same cluster, which is helpful to improve the classification accuracy and to identify the operations that remove original smartphone traces.

With the use of LibSVM, we only adopted the default kernel parameters for the linear kernel and RBF kernel, and did not make the grid search to optimize the kernel parameters. The detection performance could be improved by optimizing the kernel parameters. While using hierarchical agglomerative clustering, we only adopted standardized Euclidean distance and usual Euclidean distance to measure the pairwise distance, other distance measurements, such as Mahalanobis distance, Minkowski distance with exponent 2, correlation, and cosine distance, have not been fully examined. Additionally, our

smartphone image database is not large enough; more brands of smartphones, more smartphone images, and more operations should be included and collected in the future to conduct a more convincing examination.

To sum up, we have developed a technique that can successfully detect JPEG double compression by integrating marginal density and the neighboring joint density features in DCT domain. Our method is superior to Markov process-based approach in terms of achieving a higher detection accuracy at a lesser computational cost. Our study shows that the detection performance is related not only to the compression quality factors but also to image complexity, which is an important parameter that seems to have been so far overlooked by the research community in conducting performance evaluation. To formally study the performance evaluation issues, the image complexity and compression quality should therefore be included as a whole.

Following the success in detection of JPEG double compression, we conducted studies based on processed smartphone images to identify the smartphone source and the post-capture manipulations. Experimental results show that our method is strongly promising in correctly identifying the smartphone source and revealing the past manipulations simultaneously, including the combination of double JPEG compression, cropping, and rescale. Our studies also indicate that, due to the complexity of intentional manipulation, it is more productive to combine clustering and classification techniques together for performing the detection.


*A.3. Detection of MPEG double compression*

We conduct a qualitative statistical analysis about the impact caused by MPEG-2 compression on distributions of reconstructed DCT coefficients, and demonstrate the differences in distributions of quantized DCT coefficients between the single compression and double compression. A set of DCT distributions with different quantization scale factors are constructed to extract convex pattern features, and a novel detection algorithm is designed to detection of double MPEG compression in CBR videos. In our simulation system, the target output bit-rate, rather than quantization scale factor, is selected as the only parameter to control MPEG-2 encoders. The target output bit-rate can easily be configured through the system menu, without need to modify source codes of MPEG-2 encoders. So it makes our detection algorithm more suitable for all kinds of video coding systems, especially in some business video systems. On the other hand, our proposed detection algorithm maintains good detection performance in many cases. More specifically, it can detect double compressed videos with both high-quality and low-quality. Even if the primary compression and the secondary compression use different kinds of MPEG-2 encoders, our algorithm can also reveal the track of double MPEG-2 compression.

Future research efforts will focus on improving the detecting accuracy of our algorithm. At the same time, it is notable that detection of double MPEG compression does not

necessarily prove malicious tampering, because it is possible that a user resaves the high quality video file as a lower quality one to save storage space. As a result, we need to explore new features in the spatial domain and temporal domain of video resources, and combine existing detection techniques to reveal the real video tampering operation.

*A4. MP3 audio steganalysis*

Frequency-based subband moment statistical features provide a more accurate and stable classification than the other feature sets. More specifically, the features corresponding to the middle part of 576 subbands are more significant than those corresponding to the high and the low parts. The reason for this phenomenon is that the high and the low parts of the subbands usually contain quantized MDCT coefficients of large and small values. The MP3Stego algorithm embeds data by modifying the quantization step-size, which affects all MDCT coefficients in the particular frame. However, the effects of different values vary greatly due to the non-uniform scale of quantization. The larger values usually refer to the more informative content of the audio signal, and they are mainly determined by the characteristic of the signal. The information-hiding behavior and the high complexity of the signal could both indicate the inconsistency of the coefficients between adjacent frames. The zero value usually does not change by modifying the quantization step-size. Therefore, the features extracted from the middle frequency are more sensitive to information-hiding.

In this study, we focus on detecting the information-hiding of MP3Stego, because MP3Stego implements a unique hiding scheme which is involved in the compression process. However, the proposed approach also has the generalization capability to steganalyze other steganographic systems, because the traditional steganography introduces more distortion than MP3Stego in compressed audio.

The GGD shape parameter is introduced as an important signal complexity measure to evaluate the detection performance. With same modification density, the detection accuracy decreases as the signal complexity increases. However, the GGD shape parameter only describes the distribution of the MDCT coefficients of the entire audio and neglects the relation between MDCT coefficients in one frame or one subband. Since different complexities may have similar distributions, an accurate measure of signal complexity with fine granularity is another important issue to MP3 audio steganalysis. Since we extract the shape parameter from quantized MDCT coefficients, the distribution of these coefficients is not only influenced by the signal complexity but also by the setting and the implementation of MP3 encoder. The MP3Stego has a tight coupling between the hiding algorithm and the MP3 encoder. Although the implementations of the MP3 encoder have to comply with ISO standards, the differences in the quantization function and distortion control will affect the performance of steganalysis. The differences in the quantization function may affect the distribution of MDCT coefficients and increase the false alarm rate with a trained model using another encoder.

In summary, we propose a comprehensive approach to steganalysis of MP3 audio by

deriving a combination of features from quantized MDCT coefficients. We extract frequency-based moment statistical features, accumulative Markov transition features, and accumulative neighboring joint density features. We also model the distortion by extracting the distribution parameters of generalized Gaussian density from individual frames in the MDCT transform domain. Different feature selection algorithms are applied to improve detection accuracy. For an accurate evaluation, signal complexity and modification density are introduced to evaluate the performance of the proposed approach. Experimental results show that our approach successfully detects the information-hiding in MP3 steganograms generated by the MP3Stego steganographic tool. The proposed approach obtains reliable performance under each category of signal complexity, especially for audio signals with high signal complexity, and thus improves the state of the art of audio steganalysis.

*A5. AAC audio forgery detection*

Although our experiments presented above do not examine all possibility of AAC audio forgery due to the very high computational cost (it is very time-consuming to examine of all possible forgery shifted by 1023 positions), by simulating part cases of AAC audio forgery at a reasonable computational cost, our experimental results do verify the effectiveness of our proposed shift-recompression-based approach to detection of AAC audio forgery of the same bit rate. The detection accuracy is very promising.

It is worth noting that shift-recompression-based method is effective not only for detecting AAC audio forgery, but also for detecting MP3 audio forgery of the same bit rate.

## B.    Implications for policy and practice

Multimedia forensics is a multiple-disciplinary research field with important impacts to law enforcement. In multimedia forensics, steganography detection or steganalysis and forgery detection are two spots. It is know that that steganography had been employed by a foreign government intelligence agency and digital multimedia data can now be easily forged. Multimedia steganography and forgery techniques and the proliferation have made big challenges to law enforcement.

By exploring the characteristic modification caused by digital multimedia steganography and forgery, developing new detection algorithms/approaches, and adopting new measurement parameters for the evaluation, we have successfully achieved the project goals including "discovering the characteristic modification caused by digital multimedia steganography and forgery, developing more accurate and more reliable methods for steganalysis and digital evidence authentication, and developing a complete evaluation procedure for gaining full understanding of the accuracy, reliability, and measurement validity of steganography detection and digital evidence authentication in digital image, audio, and video files."

The implications for policy and practice lie in the following:

1.  Although multimedia steganography and forgery have made big challenges to law enforcement in protection of public safety and national security, our study shows that some advanced steganography and forgery systems can be accurately detected and hence the relevant crimes may be defeated and/or prevented. For example, in detecting several types of JPEG steganography at the relatively high information hiding ratio, our approach has achieved 100% mean accuracy over 100 experiments. Our forgery detection approaches in this study are also very promising with high detection accuracies. Therefore, we would recommend utilize the state-of-art for steganography and forgery detection for forensics purposes.

2.  The complete evaluation in multimedia forensics may include multiple parameters including information hiding ratio and/or forgery size, compression factors, hiding algorithms, multimedia signal complexity, detection algorithms, feature selection methods, classification models and learning classifiers.

3.  Our study also indicates that it is still hard to defeat some meticulously designed steganography, e.g., the data hiding takes place in the high complexity components in the multimedia signal (Liu Sung Chen and Huang 2011).

4.  The study in multimedia forensics is normally subjected to relatively simple environment with a certain knowledge and limitation to the testing multimedia data. For example, to detect some type of steganography by using a steganographic algorithm x, the steganograms are denoted as Sx, covers are denoted as C. Classification models are constructed to discriminate the steganogram from cover. It is clear that the detection is conducted in the environment that contains only Sx and C, and the outcome can be predicted either Sx or C, relatively predestinated.

    Unfortunately, the real life detection generally faces an open and complicated environment. For example, we are given a JPEG image to determine whether it is carrying a covert message or not. How do we cope with it? We cannot simply adopt the classification model that was used to distinguish Sx and C, since we are not sure about the image under examination is either an untouched cover or the type Sx of steganogram, or some other type of steganogram, or a cover that was processed by some legitimate operations.

5.  It is known that that steganography had been employed by a foreign government intelligence agency (Web justice 1; Web justice 2), and the potential usage of

steganography to disseminate covert message in social media such as on Facebook could be enormous (web secretbook). The further study in multimedia forensics is highly needed for forensics purposes.

## C.    Implications for further research

The continuous improvement of the state of steganalysis and forgery detection should be emphasized in the future study. Additionally, several new steganographic systems have been proposed to hide data in JPEG images (Liu, Sung, Chen and Huang 2011; Holub and Fridrich 2013), a Google Chrome extension is currently available to hide data in the photos by Facebook (web secretbook). AAC audio streams (Wei, Li and Wang 2010), and VoIP audio stream (Hamdaqa and Tahvildari 2011; Mazurczyk 2012) including Skype-based steganography (Mazurczyk, Karas and Szczypiorski 2013), and no effective detection methods area available to this date, which is worthy for the further exploration.

While we have designed several effective detection approaches within the state-of-the-art, the realistic detection toolkits may be implemented for the testing and validation for forensics purposes.

It is worthy of making the contribution for real life detection that generally faces an open and complicated environment. Further study may be also highlighted on revealing the processing history of the multimedia data under the examination.

# V.    References

Alles EJ, Geradts JMH and Veenman CJ (2009), "Source camera identification for heavily JPEG compressed low resolution still images", *Journal of Forensic Sciences*, 54(3): 628-638.

Avidan S and Shamir A (2007). "Seam carving for content-aware image resizing". *ACM Transactions on Graphics*, 26(3), Article 10, 2007.

Bayram S, Dirik AE, Sencar HT and Memon ND (2010). "An ensemble of classifiers approach to steganalysis", in *Proc of International Conference on Pattern Recognition*, pp. 4376-4379.

Bayram S, Sencar HT and Memon N (2005). "Source camera identification based on CFA interpolation", *Proc. IEEE Int. Conf. Image Processing*, Genova, Italy, 2005, pp. 69–72.

Bayram S, Sencar HT and Memon N (2006). "Improvements on source camera model identification based on CFA interpolation". *Proc. IFIP WG 11.9 Int. Conf. Digital Forensics*, Orlando, FL, 2006, pp. 24–27.

Bayram S, Sencar HT, Memon ND (2008). "Video copy detection based on source device characteristics: a complementary approach to content-based methods". *Proc. Int. ACM Conf. Multimedia Inf. Retr. MIR*. (Vancouver, BC, Canada, 2008), pp. 435-442.

Bayram S, Sencar HT and Memon N (2009). "An efficient and robust method for detecting copy-move forgery". *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 19-24, 2009.

Bianchi T and Piva A (2012a). "Detection of nonaligned double JPEG compression based on integer periodicity maps", *IEEE Trans. Info. Forensics and Security*, 7(2): 842-848.

Bianchi T and Piva A (2012b). "Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts", *IEEE Trans. Info. Forensics and Security*, 7(3): 1003-1017.

Böhme R and Westfeld A (2004). "Statistical characterization of MP3 encoders for steganalysis", *Proc. of 2004 Workshop on Multimedia and Security*, pp. 25–34.

Celiktutan O, Sankur B, Avcibas I (2008), "Blind identification of source cell-phone model", *IEEE Trans. Infor. Forensics and Security*, 3(3):553-566.

Chang C and Lin CJ (2011). "LIBSVM : a library for support vector machines", *ACM Trans*. *Intell*. *Syst.Technol*, vol 2, no. 3, article 27. LIBSVM is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Chang IC, Yu JW and Chang CC (2013), "A forgery detection algorithm for exemplar-based inpainting images using multi-region relation", *Image and Vision Computing*, 31(1):57-71.

Chen B and Wornell GW (2001). "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding", *IEEE Transaction on Information Theory*, vol. 47, no. 4, pp. 1423-1443.

Chen C and Shi Y (2008). "JPEG image steganalysis utilizing both intrablock and interblock correlations", In *Proc. 2008 IEEE International Symposium on Circuits and Systems*, pp. 3029–3032.

Chen C, Shi Y and Su W (2008). A machine learning based scheme for double JPEG compression detection. *Proc. of 19th ICPR*: 1-4, 2008.

Chen M, Fridrich J, Goljan M, and Lukáš J (2008). "Determining image origin and integrity using sensor noise", *IEEE Trans. Info. Forensics and Security*, 3(1) (2008), 74-90.

Chen M, Fridrich J, Lukas J and Goljan M (2007). "Imaging sensor noise as digital X-ray for revealing forgeries". *Proc. of 9th Information Hiding Workshop*. Saint Malo, France, June 2007.

Chen W and Shi Y (2008). "Detection of double MPEG compression based on first digit statistics", *Lect. Notes Comput. Sci* (Busan, Republic of Korea, 2008), pp. 16-30.

Chen W, Shi YQ and Su W (2007). "Image splicing detection using 2-D phase congruency and statistical moments of characteristic function". *Proc. SPIE*, Vol. 6505, 65050R (2007); DOI:10.1117/12.704321.

Chen Y and Hsu C (2011). "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection". *IEEE Transactions on Information Forensics and Security*, 6(2): 396-406.

Cho D and Bui TD (2005). "Multivariate statistical modeling for image denoising using wavelet transforms", *Signal Processing: Image Communication,* 20(1): 77-89.

Choi KS, Lam EY, Wong KKY(2006), "Source camera identification using footprints from lens aberration" *Proc. of the SPIE*, vol. 6069, pp. 172-179.

Ding W and Liu B (1996). "Rate control of MPEG video coding and recording by rate-quantization modeling", *IEEE Trans. Circuits and Systems for Video Technology*, 6(1): 12-20.

Dirik AE, Sencar HT, Memon N (2007), "Source camera identification based on sensor dust characteristics" *Proc IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, 2007, pages 1-6.

Fan RE, Chang KW, Hsieh CJ, Wang XR and Lin CJ (2008). "LIBLINEAR: A library for large linear classification", *Journal of Machine Learning Research,* vol. 9, pp. 1871-1874.

Farid H (1999). "Detecting digital forgeries using bispectral analysis". AI Lab, Massachusetts Institute of Technology, Tech. Rep. AIM-1657, 1999.

Farid H (2006). "Digital image ballistics from JPEG quantization". Dept. Comput.Sci., Dartmouth College, Tech. Rep. TR2006-583, 2006.

Farid H (2009), " Image forgery detection, a survey". *IEEE Signal Processing Magazine*, March 2009, 16-25.

Filler T and Fridrich J (2010). "Gibbs construction in steganography", *IEEE Trans. on Info. Forensics and Security*, 5(4): 705-720.

Filler T and Fridrich J (2011). "Design of adaptive steganographic schemes for digital images", in *Proc. SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics XIII*, San Francisco, CA, January 23-26, 2011. The hiding tool is available at http://dde.binghamton.edu/download/stego_design/

Filler T, Judas J and Fridrich J (2011). "Minimizing additive distortion in steganography using syndrome-trellis codes", *IEEE Trans. on Info. Forensics and Security*, 6(3): 920-935.

Fillion C and Sharma G (2010). "Detecting content adaptive scaling of images for forensics applications". P*roceedings of SPIE, Media Forensics and Security II*, vol. 7541, doi: 10.1117/12.838647.

Fridrich J (2005). "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes", In *Proc Information Hiding Workshop, Lecture Notes in Computer Science* vol 3200, pp.67-81.

Fridrich J, Kodovsky J, Holub V and Goljan M (2011a). "Breaking HUGO – the process discovery". In *Proc. 13th Information Hiding Workshop*, pp. 85-101. Prague, Czech Republic, May 18–20, 2011.

Fridrich J, Kodovsky J, Holub V and Goljan M (2011b). "Steganalysis of content-adaptive steganography in spatial domain", In *Proc. 13th Information Hiding Workshop*, p. 102-117, Prague, Czech Republic, May 18–20, 2011.

Fridrich J and Kodovsky J (2012). "Rich models for steganalysis of digital images", *IEEE Transactions on Information Forensics and Security*, 7(3): 868-882, 2012.

Fridrich J, Soukal D, and Lukáš J (2003). "Detection of copy-move forgery in digital images", *Proc. Digital Forensic Research Workshop*. (Cleveland, OH, 2003).

Fu D, Shi Y and Su Q (2007). "A generalized Benford's law for JPEG coefficients and its applications in image forensics". *Proc. SPIE Electronic Imaging, Security andWatermarking of Multimedia Contents IX*, vol. 6505, pp. 1L1–1L11.

Fu D, Shi Y, Zou D and Xuan G (2006). "JPEG steganalysis using empirical transition matrix in block DCT domain", In *Proc. IEEE 8th Workshop on Multimedia Signal Processin*g, pp 310-313.

Gallagher AC (2005). "Detection of linear and cubic interpolation in jpeg compressed images". *Proc. 2nd Canadian Conf. Computer and Robot Vision*, vol. 171, pp. 65–72.

Geetha S, Ishwarya N, and Kamaraj N (2010). "Evolving decision tree rule based system for audio stego anomalies detection based on Hausdorff distance statistics", *Information Sciences*, 180(13): 2540–2559.

Gou H, Swaminathan A and Wu M (2007a). "Robust scanner identification based on noise features". *Proc. of SPIE*, vol. 6505, February 2007.

Gou H, Swaminathan A and Wu M (2007b). "Noise features for image tampering detection and steganalysis". *Proc. of IEEE Int. Conf. on Image Processing* (ICIP'07), San Antonio, TX, Sept. 2007.

Gou H, Swaminathan A, and Wu M (2009). "Intrinsic sensor noise features for forensic analysis on scanners and scanned images", *IEEE Trans. Info. Forensics and Security*, 4(3): 476-491.

Gul G and Avcibas I (2009), "Source cell phone camera identification based on signular value decomposition", *Proc. 1st IEEE Inernational Workshop on Informaiton Forensics and Security*, pages 171-175.

Gul G and Kurugollu F (2011). "A new methodology in steganalysis: breaking highly undetectable steganography (HUGO)", In *Proc. 13th Information Hiding Workshop*, pp. 71-84, Prague, Czech Republic, May 18–20, 2011.

Guyon I, Weston J, Barnhill S and Vapnik VN (2002). "Gene selection for cancer classification using support vector machines", *Machine Learning* vol. 46, no. 1-3, pp. 389-422.

Hamdaqa M and Tahvildari L (2011). "ReLACK: A reliable VoIP steganography approach", Proc. 2011 5$^{th}$ International Confernce on Secure Software Integration and Reliability Improvement, pages 189-197.

Heijden F, Duin R, Ridder D and Tax D (2004). *Classification, Parameter Estimation and State Estimation-An Engineering Approach Using Matlab*, John Wiley & Sons, ISBN 0470090138, 2004. PRtools toolbox is available at: http://www.prtools.org

Hetzl S and Mutzel P (2005). "A graph-theoretic approach to steganography", In *Pro. 9$^{th}$ IFIP TC-6 TC-11 International Conference on Communication and Multimedia Security*, vol. 3677, pp. 119–128.

Holub V and Fridrich J (2013). "Digital Image Steganography Using Universal Distortion". In *Proc. 1$^{st}$ ACM Workshop on Information Hiding and Multimedia Security*, June 2013.

Hsu YF and Chang SF (2007). "Image splicing detection using camera response function consistency and automatic segmentation". Proc. 2007 IEEE International Conference on Multimedia and Expo, pp. 28-31.

Huang J and Mumford D (1999). "Statistics of natural images and models", In *Proceedings of Computer Vision and Pattern Recognition*, vol. 1, 1999.

Joachims T (2002). "Estimating the generalization performance of a SVM efficiently", in *Proc. 17$^{th}$ ICML,* pp. 431-438, 2002. *SVM_light* is available at http://svmlight.joachims.org/

Johnson MK, Lyu S, and Farid H (2005). "Steganalysis of recorded speech". *Proceedings of SPIE*, vol. 5681, pp. 664–672, 2005.

Johnson MK and Farid H (2005). "Exposing digital forgeries by detecting inconsistencies in lighting". *Proc. 7$^{th}$ ACM Multimedia and Security Workshop*, pp. 1–10.

Johnson MK and Farid H (2006). "Exposing digital forgeries through chromatic aberration". *Proc. 8$^{th}$ ACM Multimedia and Security Workshop*, pp. 48–55.

Johnson MK and Farid H (2007a). "Exposing digital forgeries through specular highlights on the eye". *Proc. 9$^{th}$ Int. Workshop on Information Hiding*, pp. 311–325.

Johnson MK and Farid H (2007b). "Exposing digital forgeries in complex lighting environments". IEEE Trans. Inform. Forensics Security, 3(2):450– 461.

Johnson MK and Farid H (2007c). "Detecting photographic composites of people". *Proc. 6th Int. Workshop on Digital Watermarking*, Guangzhou, China, 2007.

Ker A (2004). "Improved detection of LSB steganography in grayscale images", in *Proc. 6th Information Hiding Workshop*, *LNCS,* vol. 3200, pp. 97–115.

Kharrazi M, Sencar HT, Memon N (2004), "Blind source camera idetification," *Proc. ICIP'04*,Singapore, October 24-27,2004.

Kharrazi M, Sencar HT and Memon ND (2006). "Improving steganalysis by fusion techniques: a case study with image steganography", *LNCS Trans. Data Hiding and Multimedia Security* I, pp. 123-137.

Kobayashi M, Okabe T, and Sato Y (2010). "Detecting forgery from static-scene video based on inconsistency in noise level functions", *IEEE Trans. Info. Forensics and Security*, 5(4): 883-892.

Kodovsky J and Fridrich J (2009). "Calibration revisited", In *Proc. 11th ACM Multimedia and Security Workshop*, pp. 63-74. Princeton, NJ, September 7-8, 2009.

Kodovsky J, Pevny T and Fridrich J (2010). "Modern steganalysis can detect YASS", *Proc. SPIE, Electronic Imaging, Media Forensics and Security XII*, San Jose, CA, January 17–21, pp. 02-01 - 02-11, 2010.

Kodovsky J and Fridrich J (2011). "Steganalysis in high dimensions: fusing classifiers built on random subspaces", *Proc. SPIE* 7880, 78800L, 2011; doi:10.1117/12.872279

Kodovsky J, Fridrich J and Holub V (2012). "Ensemble classifiers for steganalysis of digital media", *IEEE Transactions on Information Forensics and Security*, **7**(2):432-444.

Kodovsky J and Fridrich J (2012). "Steganalysis of JPEG Images Using Rich Models", *Proc. SPIE 8303, Media Watermarking, Security, and Forensics 2012, 83030A* (February 9, 2012); doi:10.1117/12.907495.

Kurak C and McHugh J (1992). "A cautionary note on image downgrading", in: *Proceedings of the 8th Computer Security Application Conference*, pp. 153–159.

Li B, Shi Y and Huang J (2009). "Steganalysis of YASS", *IEEE Trans. Information Forensics and Security,* 4(3):369-382.

Lin Z, Wang R, Tang X, and Shum HY (2005). "Detecting doctored images using camera response normality and consistency". *Proceedings of CVPR 2005*, pp. 1087-1092.

Lin CY, Wu M, Bloom JA, Cox IJ, Miller ML and Lui YM (2001). "Rotation, scale, and translation resilient watermarking for images". *IEEE Trans. Image Processing*, vol. 10, pp. 767–782, 2001.

Liu Q, Sung AH and Ribeiro B (2005). "Statistical correlations and machine learning for steganalysis", *Proc 7th ICANNG, Adaptive and Natural Computing Algorithms*, pp. 437-440, Springer-Wien-NewYork 2005.

Liu Q, Sung AH, J. Xu and Ribeiro B (2006). "Image complexity and feature extraction for stegnalysis of LSB matching steganography", In *Proc 18th ICPR,* vol. 2, pp. 267-270, 2006

Liu Q and Sung AH (2007). "Feature mining and neuro-fuzzy inference system for steganalysis of LSB matching stegangoraphy in grayscale images", in *Proc. 20th International Joint Conference on Artificial Intelligence,* pp. 2808-2813, Jan. 2007.

Liu Q, Sung AH, Ribeiro B and Ferreira R (2008). "Steganalysis of multi-class JPEG images based on expanded Markov features and polynomial fitting", in *Proc. International Joint Conference on Neural Network 2008*, pp. 3352-3357.

Liu Q, Sung AH and Qiao M (2008). "Video steganalysis based on the expanded Markov and joint distribution on the transform domains — Detecting MSU StegoVideo", In *Proc. 7th International Conference on Machine Learning and Applications*, pp. 671-674, 2008.

Liu Q, Sung AH, Chen Z and Xu J (2008). "Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images", *Pattern Recognition,* 41(1): 56-66.

Liu Q, Sung AH, Ribeiro B, Wei M, Chen Z and Xu J (2008). "Image complexity and feature mining for steganalysis of least significant bit matching steganography", *Information Sciences,* 178(1): 21-36.

Liu Q, Sung AH, and Qiao M (2009a). "Temporal derivative-based spectrum and Mel-cepstrum audio steganalysis", *IEEE Transactions on Information Forensics and Security*, 4(3): 359–368.

Liu Q, Sung AH and Qiao M (2009b). "Improved detection and evaluation for JPEG steganalysis", in *Proc. 17th ACM Multimedia*, pp. 873-876.

Liu Q, Sung AH and Qiao M (2009c). "Novel stream mining for audio steganalysis", *Proc. 17th ACM Multimedia*, pp. 95-104.

Liu Q and Sung AH (2009), "A new approach for JPEG resize and image splicing detection", *Proc 1st ACM Workshop on Multimedia in Forensics, pp. 43-48.*

Liu Q, Sung AH, Chen Z, Liu J, Huang X and Deng Y (2009). "Feature selection and classification of MAQC-II breast cancer and multiple myeloma microarray gene expression data", *PLoS ONE*, 4(12), e8250; available at: doi:10.1371/journal.pone.0008250

Liu Q, Sung AH, Qiao M, Chen Z and Ribeiro B (2010). "An improved approach to steganalysis of JPEG images", *Information Sciences*, 180(9): 1643-1655.

Liu Q, Sung AH and Qiao M (2011a). "Neighboring joint density-based JPEG steganalysis", *ACM Trans. Intell. Syst. Technol.*, 2(2): article 16.

Liu Q, Sung AH and Qiao M (2011b). "Derivative-based audio steganalysis", *ACM Trans. Multimedia Comput. Commun. Appl,* 7(3), article 18.

Liu Q, Sung AH and Qiao M (2011c). "A method to detect JPEG-based double compression", *Proceedings of the 8th international conference on Advances in neural networks*, *Lecture Notes in Computer Science*, vol 6676, pages 466-476.

Liu Q, Sung AH, Chen Z and Huang X (2011). "A JPEG-based statistically invisible steganography", *Proc. 3rd International Conference on Internet Multimedia Computing and Service*, pp. 78-81.

Liu Q (2011a). "Steganalysis of DCT-embedding based adaptive steganography and YASS", in *Proc. 13th ACM Multimedia Workshop on Multimedia and Security*, pp. 77-86.

Liu Q (2011b). "Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery", in Proc. 3rd *ACM Workshop on Multimedia in Forensics and Intelligence*, pp. 25-30.

Liu Q, Cooper PA, Chen L, Cho H, Chen Z, Qiao M, Su Y, Wei M and Sung AH (2013). "Detection of JPEG double compression and identification of smartphone image source and post-capture manipulation", *Applied Intelligence*, 39(4): 705-726.

Liu Q and Chen L, "Improved Approaches with Calibrated Neighboring Joint Density to Steganalysis and Seam-carved Forgery in JPEG Images", *ACM Trans. Intelligent Systems and Technology*, to appear.

Lukas LJ and Fridrich J (2003). "Estimation of primary quantization matrix in double compressed JPEG images", *Digital Forensic Research Workshop* (Cleveland, OH, USA, 2003).

Lukas J, Fridrich J, and GoLjan M (2006), "Didital camera identification from sensor noise*," IEEE Transaction on Information Security and Forensics*,vol.1,pp.205-214.

Mahdian B and Saic S (2008). "Blind authentication using periodic properties of interpolation". *IEEE Trans. Inform. Forensics Security* 3(3): 529–538.

Marvel L, Boncelet C and Retter C (1999). "Spread spectrum image steganography", *IEEE Trans. Image Processing*, vol.8, no.8, pp. 1075-1083.

Mayer-Patel K, Smith B and Rowe L (2005). "The Berkeley software MPEG-1 video decoder", *ACM Trans. Multimedia Computing, Communications, and Applications,* 1(1): 110-125.

Mazurczyk W (2012). "VoIP Steganography and Its Detection - A Survey". In: Computing Research Repository (CoRR), abs/1203.4374, arXiv.org E-print Archive, Cornell University, Ithaca, NY (USA), published on 20 March 2012.

Mazurczyk W, Karas M and Szczypiorski K (2013). "SkyDe: a Skype-based steganographic method", *Int J Comput. Commun.* 8(3):432-443, June 2013.

Miche Y, Bas P, Lendasse A, Jutten C and Simula O (2009). "Reliable steganalysis using a munimum set of samples and features", EURASIP Journal on Information Security, volume 2009, article ID 901381, 13 pages. Doi: 10.1155/2009/901381.

Mielikainen J (2006). "LSB matching revisited", *IEEE Signal Processing Letters* , vol. 13, no. 5, pp. 285-287, May 2006.

Pan X and Lyu S (2010). "Region duplication detection using image feature matching". *IEEE Transactions on Information Forensics and Security*, 5(4): 857-867.

Pan X, Zhang X and Lyu S (2010). "Exposing image splicing with inconsistent local noise variances", *Proc. 2012 IEEE International Conference on Computational Photography*, April 28-29, 2012, pages 1-10.

Peng H, Long F and Ding C (2005). "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238.

Pevny T, Filler T and Bas P (2010). "Using high-dimensional image models to perform highly undetectable steganography", in *Proc. 12th Information Hiding Workshop*, pp. 161-177, Calgary, Alberta, Canada, June 28-30, 2010.

Pevny T and Fridrich J (2007). "Merging Markov and DCT features for multi-class JPEG steganalysis", in *Proc. SPIE*, Vol. 6505, 650503, 2007; DOI:10.1117/12.696774.

Pevny T and Fridrich J (2008a). "Detection of double-compression in JPEG images for applications in steganography". *IEEE Trans. Information Forensics and Security*, 3(2):247-258.

Pevny T and Fridrich J (2008b). "Multi-class detector of current steganographic methods for JPEG format", *IEEE Transactions on Information Forensics and Security*, 3(4): 635–650.

Pospescu AC and Farid H (2004a). "Exposing digital forgeries by detecting duplicated image regions", *Technical Report* (Dartmouth College, Computer Science, 2004).

Pospescu AC and Farid H (2004b). "Statistical tools for digital forensics", *Proc. 6th Int. Workshop on Information Hiding* (Toronto, Canada, 2004), pp.128 -147.

Popescu AC and Farid H (2005a). "Exposing digital forgeries by detecting traces of re-sampling". *IEEE Trans. Signal Processing* 53(2): 758–767.

Pospescu AC and Farid H (2005b). "Exposing digital forgeries in color filter array interpolated images", *IEEE Trans. Signal Processing*, 53(10): 3948-3959.

Prasad S and Ramakrishnan KR (2006). "On resampling detection and its application to image tampering". *Proc. IEEE Int. Conf. Multimedia and Exposition* 2006, pp. 1325–1328.

Provos N (2001). "Defending against statistical steganalysis", in *Proc. 10th USENIX Security Symposium*, vol.10, pp. 323-335, 2001.

Ohm JR (2004). *Multimedia Communication Technology, Representation, Transmission and Identification of Multimedia Signals*. Springer, Berlin, 2004.

Qiao M, Sung AH, and Liu Q (2009), "Steganalysis of MP3Stego", *Proceedings of 22nd International Joint Conference on Neural Networks*, pp. 2566–2571, 2009.

Qiao M, Sung AH, and Liu Q (2010a), "Predicting embedding strength in audio steganography", *Proceedings of 9th IEEE International Conference on Cognitive Informatics,* pp. 925–930, 2010.

Qiao M, Sung AH and Liu Q (2010b). *"*Revealing real quality of double compressed MP3 audio". *Proc. ACM Multimedia* 2010, pages 1011-1014.

Qiao M, Sung AH and Liu Q (2013). "MP3 audio steganalysis", *Information Sciences*, vol. 231, pp. 123-134, May 2013.

Reininger RC and Gibson JD (1983). "Distributions of the two dimensional DCT coefficients for images", *IEEE Trans. Commun*, 31(6): 835-839.

Sachnev V, Kim HJ and Zhang R (2009). "Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome code", In: *Proc. 11th ACM Multimedia & Security Workshop,* pp. 131-140, 2009.

Sallee P (2003). "Model-based steganography", in *Proc. 2003 International Workshop on Digital Watermarking,* vol. 2939, pp. 154–167, Oct. 2003.

Sallee P (2005). "Model-based methods for steganography and steganalysis", *International Journal of Image and Graphics*, vol. 5, no. 1, pp. 167-189, 2005.

Sarkar A, Nataraj L and Manjunath BS (2009). "Detection of seam carving and localization of seam insertions in digital images". *Proceedings of 11$^{th}$ ACM MM&Sec*, pages 107-116, 2009.

Sharifi K and Leon-Garcia A (1995). "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 52–56.

Shi Y, Chen C and Chen W (2006). "A Markov process based approach to effective attacking JPEG steganography", *Proc. 8$^{th}$ Information Hiding Workshop*, pp. 249-264, 2006.

Shi YQ, Chen C and Chen W (2007). "A natural image model approach to splicing detection". *Proc. 9$^{th}$ workshop on Multimedia & Security*, pp. 51-62.

Solanki K, Sarkar A and Manjunath B (2007). "YASS: Yet another steganographic scheme that resists blind steganalysis", in *Proc. 9$^{th}$ Information Hiding Workshop*, pp. 16-31, 2007.

Sun T, Wang W, and Jiang X (2012). "Exposing video forgeries by detecting MPEG double compression", *Proc. Int. IEEE Conf. Acoustics, Speech and Signal Processing* (Piscataway, NJ, USA, 2012), pp. 1389-1392.

Swaminathan A, Wu M, and Liu KJR (2008). "Digital image forensics via intrinsic fingerprints", *IEEE Trans. Info. Forensics and Security*, 3(1):101-117.

Tsai MJ and Wu GH (2006), "Using image features to identify camera sources", *Proc. IEEE ICASSP* 2006, May 14-19, 2006.

Tsai MJ, Lai CL and Liu J (2007), "Camera/mobile phone source identification for digital forensics", *Proc. ICASSP* II-221-II-224, April 15-20, 2007.

Vapnik VN (1998). *Statistical Learning Theory*, John Wiley, 1998.

Wang L (2000). "Rate control for MPEG video coding", *Signal Processing: Image Communication*, 15(6): 493-511.

Wang W and Farid H (2006). "Exposing digital forgeries in video by detecting double MPEG compression". *Proc. ACM Multimedia and Security Workshop* (Geneva, Switzerland, 2006), pp. 37-47.

Wang W and Farid H (2009). "Exposing digital forgeries in video by detecting double quantization", *Proc. ACM Multimedia and Security Workshop* (Princeton, NJ, 2009), pp. 39-47.

Wei Y, Li G and Wang Y (2010). "Controlling bitrate steganography on AAC audio", *Proc. 3$^{rd}$ International Conference on Image and Signal Processing*, pages 4373-4375/

Westfeld A (2001). "High capacity despite better steganalysis (F5 − a steganographic algorithm)", in *Proc. 4$^{th}$ Information Hiding Workshop*, pp. 289–302, 2001.

Yang R, Qu Z and Huang J (2008). Detecting digital audio forgeries by checking frame offsets. *Proc. 10th ACM Workshop on Multimedia and Security*, pages: 21-26.

Yu Y, Zhou J, Wang Y and Chen C (2001). "A novel two-pass VBR coding algorithm for fixed-size storage application", *IEEE Trans. Circuits Syst Video Technol.* 11(3): 345-356.

Zhou X and Tuck DP (2007). "MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data", *Bioinformatics,* vol. 23, no. 9, pp. 1106–1114, 2007.

ISO/IEC IS 13818-2: Information technology – generic coding of moving pictures and associated audio information- Part 2: Video, 1995.

Test Model 5 for ISO/MPEG II, Apr. 1991.

(Web Adobe Premiere Pro) Adobe Premiere Pro 2.0: A popular MPEG-2 encoder. Available at http://www.adobe.com/products/premiere/

(Web VQEG) Test Sequences, Video Quality Experts Group, Available at http://www.its.bldrdoc.gov/vqeg/downloads/downloads.php.

ISO/IEC JTC1/SC29/WG11, IS11172-3, Information Technology–Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s, Part 3: Audio, 1992.

(Web justice1) http://www.justice.gov/opa/documents/062810complaint1.pdf

(Web justice2) http://www.justice.gov/opa/documents/062810complaint2.pdf

(Web photoshop-cs4) http://www.photoshopsupport.com/photoshop-cs4/what-is-new-in-photoshop-cs4.html

(Web iresizer) http://www.iresizer.com/

(Web liquidrescale) http://liquidrescale.wikidot.com/en:examples

(Web digikam) http://www.digikam.org/node/439

(Web imagemagick) http://www.imagemagick.org/Usage/resize/#liquid-rescale

(Web boss) http://www.agents.cz/boss/BOSSFinal/

(Web theblaze) http://www.theblaze.com/stories/nkorea-caught-doctoring-photo-of-kim-jong-ils-funeral-can-you-spot-the-difference/

(Web cbsnews1) http://www.cbsnews.com/8301-503543_162-20016679-503543.html

(Web cbsnews2) http://www.cbsnews.com/stories/2010/09/17/world/main6876519.shtml

(Web latimesblogs) http://latimesblogs.latimes.com/babylonbeyond/2008/07/iran-doctored-m/comments/page/2/

(Web AAC) http://en.wikipedia.org/wiki/Advanced_Audio_Coding

(Web MP3Stego) http://www.petitcolas.net/fabien/steganography/

(Web Audiocoding) http://www.audiocoding.com/faac.html

(Web Secretbook) https://chrome.google.com/webstore/detail/secretbook/plglafijddgpenmohgiemalpcfgjjbph

# VI. Dissemination of Research Findings

**Poster Presentation**

1. Liu Q, Cooper PA and Sung AH (2011). Novel feature mining to detect adaptive steganography and forgery in JPEG images. *39th Annual ASCLD Symposium*, Denver, Colorado, on September 21, 2011.
2. Liu Q, Li X and Cooper PA (2012). A new approach to detecting seam-carved forgery in JPEG images. *2012 NIJ Conference Poster Session*, Washington DC, on June 19, 2012

**Peer-Reviewed Publications**

1. Liu Q, Sung AH and Qiao M (2011). A method to detect JPEG-based double compression. In Advances in Neural Networks–ISNN 2011, pp. 466-476. Springer Berlin/Heidelberg
2. Liu Q (2011). Steganalysis of DCT-embedding based adaptive steganography and YASS, in *Proc. 13th ACM Multimedia Workshop on Multimedia and Security*, pages 77-86.
3. Liu Q (2011). Detection of misaligned cropping and recompression with the same quantization matrix and relevant forgery, in *Proc. 3rd ACM Workshop on Multimedia in Forensics and Intelligence*, pages 25-30.
4. Liu Q, Li X, Chen L, Cho H, Cooper P, Chen Z, Qiao M and Sung AH (2012). Identification of smartphone-image source and manipulation. *Advanced Research in Applied Artificial Intelligence, Lecture Notes in Computer Science*, vol. 7345, pages 262-271.
5. Liu Q, Li X, Cooper PA and Hu X (2012). Shift recompression-based feature mining for detecting content-aware scaled forgery in JPEG images, in *Proc 12th International Workshop on Multimedia Data Mining (MDM-KDD),* pages 10-16.
6. Xu J, Su Y and Liu Q (2013). Detection of double MPEG-2 compression based on distributions of DCT coefficients. *International Journal of Pattern Recognition and Artificial Intelligence*. vol. 27(1), DOI: 10.1142/S0218001413540013
7. Qiao M, Sung AH, and Liu Q (2013). MP3 audio steganalysis, *Information Sciences*, Volume 231, 10 May 2013, Pages 123–134
8. Liu Q, Cooper PA, Chen L, Cho H, Chen Z, Qiao M, Su Y, Wei M, Sung AH(2013). Detection of JPEG double compression and identification of smartphone image source and post-capture manipulation. *Applied Intelligence*, December 2013, Volume 39, Issue 4, pp 705-726.
9. Liu Q, Cooper PA, Zhou B (2013). An improved approach to detecting content-aware scaling-based tampering in JPEG images*, Proc 2013 IEEE China Summit &*

*International Conference on Signal and Information Processing (ChinaSIP*), pp. 432-436.

10. Liu Q and Chen Z. Calibrated neighboring joint density to detect steganography and seam-carved forgery in JPEG images. *ACM Transactions on Intelligent Systems and Technology*, accepted.

**Patent**

1. Liu Q (2013). Steganalysis with neighboring joint density. Publication date: 2013-08-15, Patent application number: 20130208941, http://patents.justia.com/patent/20130208941

Investigations have also been performed on a peripheral -- but fundamentally relevant -- topic of interest to steganalysis, machine-learning based multimedia analysis and forensics, and data mining problems in general, which is the issue of determining if a "critical dimension" exist for a given combination of dataset and learning machine. Some preliminary results indicate this is a potentially fruitful direction for future research, and are published in selected international conference venues.

1. Suryakumar D, Sung AH, and Liu Q (2013), "Influence of Machine Learning vs. Ranking Algorithm on the Critical Dimension", International Journal of Future Computer and Communication, Kuala Lumpur, Malaysia, doi: 10.7763/IJFCC, vol: 2, no. 3, pp. 215 – 220, 2013

2. Suryakumar D, Mazumdar S, Sung AH, and Liu Q( 2012), "Performance Evaluation of Bio-Medical Datasets at Their Critical Dimension", 12th International Conference on Intelligent Systems Design and Applications, ISDA, Cochin, India, doi:10.1109/ISDA.2012.6416629, pp. 740 – 745, 2012

3. Suryakumar D, Sung AH, and Liu Q (2012), "Critical Dimension in Data Mining", IRAIA, International Conference on Information, Process, and Knowledge Management, eKNOW 2012, Valencia, Spain, pp. 97-101, 2012

4. Suryakumar D, Sung AH, and Liu Q (2012), "Dependency of Critical Dimension on Learning Machines and Ranking Methods", IEEE IRI, Las Vegas, Nevada, doi:10.1109/IRI.2012.6303086, pp. 738 – 739, 2012

5. Suryakumar D, Sung AH, and Liu Q (2011), "Determine the Critical dimension in data mining (experiments with bioinformatics datasets)", ISDA, Valencia, Spain, doi:10.1109/ISDA.2011.6121702, pp. 481 – 486, 2011