2013

Video Steganalysis for Digital Forensics Investigation

Kevin Bryan
University of Rhode Island, bryank@cs.uri.edu

VIDEO STEGANALYSIS FOR DIGITAL FORENSICS INVESTIGATION

BY

KEVIN BRYAN

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2013

DOCTOR OF PHILOSOPHY DISSERTATION

OF

KEVIN BRYAN

APPROVED:

Dissertation Committee:

Major Professor Victor Fay-Wolfe

Lutz Hamel

Qing Yang

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2013

ABSTRACT

Increasing use of steganography in espionage and exfiltration of company secrets means that it is important to find ways to detect such activity. Because the amount of data being transferred is also growing, channels that can hide larger amounts of data are going to become increasingly attractive. This research will focus on detecting hidden data in one such medium, namely MPEG video.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

Introduction

Steganography is the process of hiding information in "plain sight". Secret communication is possible by modifying a *cover* medium is to *embed* data. An analog example might be adding microscopic dots to an image or document. Digital steganography manipulates bits of data to embed the secret message with minimal impact on the interpretation of the original data.

Video steganography is an emerging sub-field of digital steganography. Most digital steganographic methods have relied on exploiting file formats to hide information in parts of files either not parsed or parts invisible to the user in normal processing and use. Some more advanced methods hide data in the noise produced by lossy compression formats, such as JPEG images or MP3 audio files. For example, compare the two flowers in Figure 1 and try to decide which is stegged.

Many of the JPEG image steganographic techniques carry over to MPEG video, however the steganalysis for video can be different because of the increase in the volume of data. Given the relatively high capacity of video, it is likely that it will be the next most popular carrier to discreetly transfer large amounts of data.

(a) (b)

Figure 1. Clean and stegged versions of an image

The adoption rate of steganography in general and video steganography in particular is not well known, however there have been recent accounts in the news of law enforcement finding evidence of its use on suspect machines. Since steganography scanning tools are not yet mature enough for regular use, most of the evidence comes from steganographic tools themselves being installed.

Videos provide fairly high bandwidth for data embedding and are frequently posted and transferred on-line. The goal of steganalysis is to reduce the effective bit rate of data embedding in video by reliably detecting the higher embedding rates.

This research will focus on developing methods to detect steganography in digital video. The highest capacity channel in video is changing the Discrete Cosine Transform (DCT) coefficients that are used to encode frames of video. This method is easy to implement based on adaptations of existing JPEG steganographic tools to MPEG encoders, and therefore is likely to be the most prolific type.

Chapter 2 provides background on MPEG compression, digital steganography and digital steganalysis. Chapter 3 presents the methodology that this project developed for video steganalysis. The results of this method are found in Chapter 4. Finally, concluding remarks and a discussion of the future direction of video steganalysis is presented in Chapter 5.

CHAPTER 2

Background

In order to understand the problem space, it is necessary to first describe how digital steganography works and how it is applied during compression MPEG video. The following presents only the relevant parts of the MPEG specification. Examination of how data can be hidden and how others have sought to find will follow.

2.1 MPEG Compression

Almost all lossy compression techniques that deal with perceived media exploit that human senses do not distinguish small changes in high frequency information. Visually this manifests as high detail areas of an image. In raw video every pixel is represented by 3 bytes, either by separating into red, green and blue (RGB), or, more likely, luma and two chroma components, called YUV or YCbCr. The human eye is less sensitive to color than intensity so MPEG always encodes using YUV with the chroma components down sampled by a factor of two horizontally and vertically. To aid the removal of high frequency data, MPEG uses Discrete Cosine Transform (DCT) to convert the raw (spatial) data into the frequency domain. It does this in 8×8 blocks of pixels. The upper left coefficient is called the DC coefficient and represents an average intensity across the block, and the rest are AC coefficients. The DCT coefficients that come from the conversion are then divided (quantized) by amounts weighted by their importance. For example, the lowest frequency, the DC coefficient, is always divided by 8, whereas the highest frequency component is divided by 83 in the default quantization matrix [1]. The default quantization matrix is designed to give the best trade-off in compression and image quality.

MPEG files typically have three types of video frames. These are called Intra-coded, Predictive-coded, and Bidirectionally predictive-coded, however they are usually abbreviated to I, P, or B frames. I-frames are entirely encoded using DCT values as discussed above. P-frames and B-frames are coded with reference to other I- or P-frames, as shown in the top of Figure 2. These references are in the form of motion vectors that indicate blocks to copy, as shown in the bottom of Figure 2. If it were a direct copy, this would reduce the 256 DCT values for a block to 2–4 values for the x and/or y offsets. In most cases, there is some residual change, so the difference between the copied block and the new block may be encoded as a DCT block, however, of such small energy, both in terms of value count and magnitude, that it takes less space to encode [1]. If no appropriate block is found in a reference frame, that block is converted to DCT coefficients directly.

For steganography, the DCT values in any or all of these frame types may be changed by a small amount with little distortion to the image when it is decoded. The technique is borrowed from JPEG steganography, which uses a similar DCT transform for its compression. The steganalysis techniques that are in use here are then also borrowed and adapted from the JPEG domain.

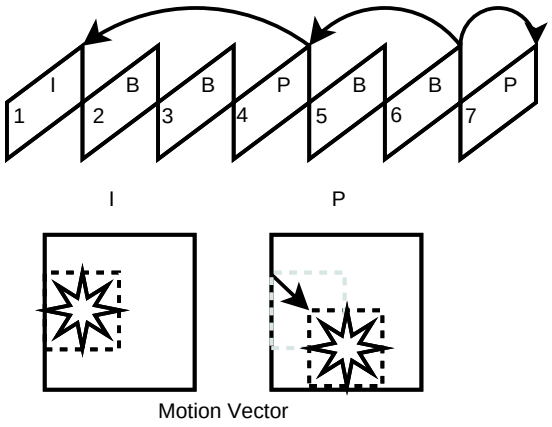

Figure 2. Frame references and motion vectors in MPEG

2.2 Digital Steganography

Digital steganography embeds information into binary data by subtly changing bits that do not have a significant effect on their interpretation. For example, consider a bitmap image file. The structure of the file is fairly simple, with a small bit of metadata at the beginning that identifies the type of file, image dimensions and color attributes, followed by the pixel data. Incorrect modification of any of the 50 byte header could make an image renderer fail or report an error. Changing the pixel data, however, will not stop it from rendering properly. Changing a single pixel by a small amount will probably not lead to a perceptible change in the image, depending on the complexity of the image. For a photograph quality picture, almost all of the values might be changed by ± 1 with no degradation [2]. This research looks at changes to the visual data, not the metadata.

For the purposes of this project, steganography and watermarking work interchangeably, however watermarking is usually more focused on perceptual invisibility and less on statistical invisibility. That is, a watermark should not change the appearance of the video, however attempting to remove the watermark should destroy the quality of the video. Detection of the existence of a watermark is usually not considered as a criterion. Most watermarks embed data that resolves to a black and white image, however that is not required.

2.2.1 Images

While digital steganography can take many forms, it has achieved most of its popularity in images. This is mostly because of the prevalence of images and their relatively high capacity. Early forms of image steganography took the form of changing the least significant bits of pixel color values[3, 4]. Pixel based least-significant-bit (LSB) steganography is unreliable when images are compressed using lossy JPEG compression, which is the most common image file type.

Steganography in JPEG images is easy to implement using libraries that give direct access to the quantized DCT coefficients. Several implementations of this exist ([5, 6, 7, 8]). Changing these values (usually by LSB bit-flipping) and then storing them back produces steganography that is much harder to detect than changing pixel values directly, despite that this change will effect a larger visual block (8×8 pixels).

2.2.2 Video

There are many stages at which data may be embedded into compressed digital video. The first stage would work in the spatial domain directly, changing the actual color values [9, 10, 11, 12, 13]. These techniques usually do not survive compression, so they require error correcting codes to ensure data fidelity. The second stage would be during frequency transform. This can be done with either Discrete Cosine Transform [14, 15, 16, 17, 18, 19, 20] or Discrete Wavelet Transform [21]. There is also an example in [20] of changing the value of motion vectors to embed data, although this will generally have a lower bit-rate. Finally, in [22], they change Huffman code pairs for AC coefficients in the bit stream directly. This has the advantage of being fast because the video is not fully decoded to images.

Watermarking techniques such as [9] are designed to withstand noise attacks by embedding data throughout several bit planes of the image. Bit planes are single bit cross-sections of an image which provide different levels of detail. Most steganographic techniques will only modify the least significant bit, or the lowest numbered bit-plane. By modifying higher bit planes (up to the fourth bit plane), [9] assures that trying to destroy the watermark also destroys the video. As in most watermarking techniques, data fidelity is not significant as long as strong correlation with the known watermark signal is found.

In [10] the goal is to avoid detection of the watermark, as well as provide

resistance. By using only embedding the watermark in several small regions within a frame, and consistently identifying those regions across frames in a scene, they provide resistance against cropping and row/column deletion.

Although Discrete Wave Transform(DWT) is similar to DCT, since it is a type of frequency transform, data embedded using DWT will go back into the spatial domain before the video compression starts. Because it does not necessarily use the same block size as DCT, data embedded via DWT can be more robust than either DCT or spatial embeddings. This is shown in [13] by performing watermarking in several frequency bands and showing the results of various attacks. The other pre-compression DWT example in [12] uses the high frequency bands to determine where embedding in the low frequency bands will have the least impact.

Although not widely used, Motion-JPEG2000 uses DWT for video compression. In [21], they look at changing the bits of the DWT coefficients based on a complexity metric.

Since MPEGs also contain motion vector information, which provides sub-pixel resolution in block copies, there is room for some data to be embedded there without disturbing the image quality. This is the technique used in [20]. Since only P and B frames contain motion vectors, they also use DCT embedding in the I frame for control information needed by their algorithm.

By analyzing a given MPEG video, [22] finds unused variable-length-code (VLC) pairs that are used in the lossless compression phase. By using a "key" of these unused pairs, and modifying existing VLCs in the compressed stream, data can be embedded easily. This method has a very low bit-rate and can sometimes have a large key size.

In the category this project is focusing on, that of DCT embedding, there are several watermarking techniques, but no steganographic implementations. In

[16] they embed data in the mid-range frequency coefficients where the complexity of the data embedded is less than the block it is being inserted into. Removing high-frequency coefficients to produce a pattern across blocks is employed in [17]. Adapting the embedding rate based on quality and frame type is done in [18]. An MPEG-4 based implementation is offered in [14] where testing shows that data embedding is visually noticeable in low bit rate videos.

2.3 Digital Steganalysis

This section is broken into two pieces. The first explains some of the work done with static images, which has direct applicability to video in that they use similar encodings. The second piece explains the work that has been done with video.

2.3.1 Images

While pixel-based LSB embedding is easy to implement and visually undetectable if there are no solid color areas in the image, it is also fairly easy to detect by simple statistics [23].

Most current work looks at DCT embedded steganography. Early attempts to detect this type of steganography using techniques such as image quality metrics [2] and wavelets [24, 25]. There have been also been many approaches to detecting DCT-based steganography[26, 27, 28]. However, in [29] Fridrich achieves the best performance to date, with percent accuracies in the high nineties for some steganographic programs at 25% of the maximum embedding rate. There are two reasons this technique works so well. The first is by using a reference image for "calibration". In "blind" steganalysis, the original non-stegged image is not available for comparison. By slightly cropping the suspect image, it is possible to create an image that is similar enough to make a good approximation. The approximation

is then used as a comparison for the statistics from the original. The second reason for increased accuracy is from focusing on exactly the data is being changed. It is specifically looking at the distribution of DCT values both between blocks and within blocks, whereas other the techniques were looking at less specific metrics, in hopes of detecting many types of steganography. It is this set of features that will form the basis of the work here, as the focus is on DCT encoded embeddings in MPEG.

Previous work

The URI research group has done previous work[30] in the area of image steganalysis by evaluating some feature sets, starting with Farid's work with Wavelets [31] and ending with Fridrich's work in [29]. Fridrich's feature sets showed the best accuracy, and were then evaluated under varying conditions. In particular, whereas previously the models were built with images against one quality level and then only tested against images of the same quality, our group evaluated using fewer models that spanned ranges of quality. Ranges of 10% image quality gave sufficiently good results, and then images of quality less than 50% are easily classified by the 50%-59% range model. Thus only 5 models need to be built, instead of 100. In addition, a model trained with a low embedding rate did well in detecting images with higher embedding rates, which gives a further reduction in the number of models necessary.

2.3.2 Video

For video steganalysis, an early but comprehensive treatment is from Budhia [32]. This work looked at detecting data embedded using additive white Gaussian noise in the spatial domain. By using data from surrounding frames, which they call *collusion*, an estimation of the current frame is achieved. Several different

collusion approaches are tried, including simple linear averaging, weighted averaging and block based reconstruction of reference frames. Block based reconstruction searches for similar blocks in nearby frames and copies them into a new reference frame. The difference of this reference frame and the original is then used to estimate the embedded data. Their features use statistics such as kurtosis, entropy, and 25^{th} percentile over this estimation. They mention that their technique can apply to the DCT domain and test it using two different methods of embedding, though without considering the encoding process (for example, P/B frames).

A performance enhancement on [32] is proposed by Jainsky in *MoViSteg* [33] which also uses motion estimation to reconstruct a frame. They employ an asymptotic relative efficiency based detector, which "is efficient for large samples and weak signals" [33]. The detector uses an adaptive threshold that is based on statistics from sample frames in the video. While they do not give overall accuracy, they report at 60% true positive to 10% false positive rate at 75 dB Peak Signal-to-Noise-Ratio (PSNR).

Most recently in [34], B. and F. Liu use collusion with a window of frames limited by a predetermined correlation threshold. They use a simple linear collusion that averages the surrounding frames. While they obtain good results (from 88–100% at 40% embedding, depending on the embedding scheme), the watermarking techniques they test against make very distinctive changes in the DCT values used. Two of them increase the range of values, which will show up in the global histogram. Another simply removes several DCT values in select blocks, which would cause noise in the dual histogram.

2.3.3 Tools

Most of the implementation of this project used *MATLAB* [35] for manipulation of frame data. Because of its natural ability to deal with multi-dimensional

data, manipulating and gathering data is easily expressed in its programming language. For example, $sum(I(:) \sim = 0)$ expresses the number of non-zero elements in an array, regardless of the number of dimensions. Additionally, some of the existing image steganography and steganalysis work was available in *MATLAB* form. MPEG encoder [36] and decoder[37] libraries were also available as *C* extensions to *MATLAB*.

The frame classifiers used the Linear Discriminant Analysis implementation from the statistical programming language *R* [38]. *R* was able to import the features extracted, build and evaluate models, and classify the frames. Other scripts converted the frame classifications into video classifications.

List of References

[1] ISO 11172-2:2003, *ISO 11172, Part 2: Video*. ISO, Geneve, Switzerland, 2003.

[2] I. Avcibaş, N. Memon, and B. Sankur, "Steganalysis using image quality metrics." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 12, no. 2, pp. 221–9, Jan. 2003.

[3] M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding for images," in *Information Hiding*. Springer, 2001, pp. 27–41.

[4] L. Marvel, C. Boncelet Jr, and C. Retter, "Spread spectrum image steganography," *Image Processing, IEEE Transactions on*, vol. 8, no. 8, pp. 1075–1083, 2002.

[5] N. Provos, "Defending against statistical steganalysis," in *10th USENIX Security Symposium*, vol. 10. Citeseer, 2001, pp. 323–336.

[6] S. Hetzl and P. Mutzel, "A graph–theoretic approach to steganography," in *Communications and Multimedia Security*. Springer, 2005, pp. 119–128.

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

[8] K. Solanki, A. Sarkar, and B. Manjunath, "YASS: Yet another steganographic scheme that resists blind steganalysis," in *Proceedings of the 9th international conference on Information hiding*. Springer-Verlag, 2007, pp. 16–31.

[9] B. Mobasseri, "Direct sequence watermarking of digital video using m-frames," *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, pp. 399–403, 1998.

[10] K. Su, D. Kundur, and D. Hatzinakos, "Spatially localized image-dependent watermarking for statistical invisibility and collusion resistance," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 52–66, Feb. 2005.

[11] D. Vatolin and O. Petrov, "MSU StegoVideo," 2007.

[12] C. Xu and X. Ping, "A Steganographic Algorithm in Uncompressed Video Sequence Based on Difference between Adjacent Frames," *Fourth International Conference on Image and Graphics (ICIG 2007)*, pp. 297–302, Aug. 2007.

[13] A. Eskicioglu and E. Elbasi, "Robust DWT Based MPEG-1 Watermarking in Four Bands," *The Second Secure Knowledge Management Workshop (SKM), New York City, NY*, 2006.

[14] A. Gupta and P. Gupta, "Digital Watermarking of MPEG-4 Videos," Indian Institute of Technology Kanpur, Tech. Rep., 2003.

[15] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, May 1998.

[16] C.-T. Hsu and J.-L. Wu, "DCT-Based Watermarking for Video," *IEEE Transaction on Consumer Electronics*, vol. 44, no. 1, pp. 206–216, Jan. 1998.

[17] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, no. 1, pp. 148–58, Jan. 2001.

[18] A. Sarkar, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Adaptive MPEG-2 video data hiding scheme," *Proceedings of SPIE*, pp. 65 051D–65 051D–9, 2007.

[19] D. Simitopoulos, S. a. Tsaftaris, N. V. Boulgouris, A. Briassouli, and M. G. Strintzis, "Fast Watermarking of MPEG-1/2 Streams Using Compressed-Domain Perceptual Embedding and a Generalized Correlator Detector," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 8, pp. 1088–1106, 2004.

[20] C. Xu, "Steganography in Compressed Video Stream," *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, pp. 269–272, 2006.

[21] H. Noda, T. Furuta, M. Niimi, and E. Kawaguchi, "Application of BPCS Steganography to Wavelet Compressed Video," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, no. 1, 2004, pp. 2147–2150.

[22] B. G. Mobasseri and M. P. Marcinak, "Watermarking of MPEG-2 video in compressed domain using VLC mapping," *Proceedings of the 7th workshop on Multimedia and security - MM&Sec '05*, pp. 91—-94, 2005.

[23] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," *Proceedings of the 2001 workshop on Multimedia and security new challenges - MM&Sec '01*, p. 27, 2001.

[24] S. Lyu, D. Rockmore, and H. Farid, "A digital technique for art authentication." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 49, pp. 17 006–10, Dec. 2004.

[25] S. Lyu and H. Farid, "Steganalysis Using Higher-Order Image Statistics," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[26] T. Pevný and J. Fridrich, "Towards multi-class blind steganalyzer for JPEG images," *Digital Watermarking*, pp. 39–53, 2005.

[27] D. Fu, Y. Shi, D. Zou, and G. Xuan, "JPEG Steganalysis Using Empirical Transition Matrix in Block DCT Domain," *2006 IEEE Workshop on Multimedia Signal Processing*, pp. 310–313, Oct. 2006.

[28] Y. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *Information Hiding*. Springer, 2007, pp. 249–264.

[29] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," *Proceedings of SPIE*, pp. 650 503–650 503–13, 2007.

[30] N. R. Bennett, "JPEG Steganalysis & TCP/IP Steganography," Ph.D. dissertation, University of Rhode Island, 2009.

[31] H. Farid and S. Lyu, "Higher-order Wavelet Statistics and their Application to Digital Forensics 2 . Image Statistics," *IEEE Workshop on Statistical Analysis in Computer Vision*, 2003.

[32] U. Budhia and D. Kundur, "Digital video steganalysis exploiting collusion sensitivity," *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, vol. 5403, pp. 210—-221, 2004.

[33] J. S. Jainsky, D. Kundur, and D. R. Halverson, "Towards digital video steganalysis using asymptotic memoryless detection," *Proceedings of the 9th workshop on Multimedia & security - MM&Sec '07*, p. 161, 2007.

[34] B. Liu, F. Liu, and P. Wang, "Inter-frame Correlation Based Compressed Video Steganalysis," *2008 Congress on Image and Signal Processing*, pp. 42–46, 2008.

[35] "Matlab." [Online]. Available: http://mathworks.com/products/matlab/

[36] D. Foti, "mpgwrite," Last retrieved June 2012, 1999. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/309

[37] D. Foti, "mpgread," Last retrieved June 2012, 1999. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/308-mpgread

[38] "R." [Online]. Available: http://www.r-project.org/

CHAPTER 3

Methodology

The goal of this project, to classify videos as stegged or not stegged, requires several steps. The first is to convert raw videos into MPEG. Raw videos must be used to ensure that the stream has not been re-compressed, which can confuse the steganalysis[1]. The next step embeds data into the MPEG files to create the stegged files. Steganalysis then extracts data from the stegged files to create features for the statistical classifier. Because the steganalysis methods are derived from image steganalysis, the features are output on a per frame basis. The statistical classifier will then make the decision on whether each frame is stegged. Finally, the video steganalysis looks at some or all of the frames to determine if the entire video is stegged.

3.1 Video Steganography

Before video steganalysis could begin, a prototype video steganography tool needed to be developed. Although there is at least one freely available tool that embeds data into video, it does so at the pixel level, before compression. Because the data is embedded before compression, much of the data could be lost during compression. To get around this requires repeated embedding with error correcting codes, which significantly reduces the capacity of the channel. Embedding after compression makes this unnecessary unless the goal is to outwit an "active warden" that manipulates the video, by re-compression, for example.

As the techniques under evaluation for steganalysis in this project are based on DCT modification, the first step of the project was to use techniques for image steganalysis and modify them to work frame-by-frame on video. A version of the reference MPEG encoder [2] and decoder[3] designed for MATLAB is used for this

purpose. Modifications to these MATLAB routines were made to extract the DCT values into MATLAB arrays during decoding, and to allow those modified arrays to be used during encoding. The arrays are modified in MATLAB. The algorithms used are described in the next section. The modified coefficients are then written to a new file that is otherwise identical to the cover video.

Normally during compression of blocks of P and B frames, the difference between the re-rendered block from the reference frame and the original is computed. The decision to use intra-coding, i.e., DCT values, is based on the variance of this difference for the block. Because embedding data may actually decrease this variance, it is possible that the modification could create DCT values that, if tested, would suggest it should not have been encoded. A good MPEG steganographic program would avoid making these types of changes. Analysis of P and B frames for most videos suggests that they do not hold as much data, and so might not be worth using anyway.

Data generation

The stegged videos used for analysis were created using all combinations of two algorithms, two video qualities, and five embedding rates. The video qualities are normal (qscale=2) and low quality (qscale=5). The embedding rates were chosen to match previous work done with images for comparison. The following is a summary of those parameters, and the description follows:

steg algorithm Our SimpleSteg routines, as well as a modified MBSteg were used to embed data

qscale A quality parameter used during MPEG encoding. Values tested were 2 and 5.

embedding rate Percentage of DCT coefficients changed. Rates tested were: 5,

10, 15, 20, 50

Two steganographic algorithms were implemented for testing the steganalysis techniques against. *SimpleSteg* is our own routine that changes all AC coefficients with $abs(x) > 1$ until the embedding rate is reached. The other steganography algorithm is version of MBSteg[4] altered slightly to work with video data. MBSteg attempts to keep the global and individual histograms unchanged as well as remove the "blockiness" that can occur when changing DCT values.

The qscale parameter changes the coarseness of the quantization done during encoding. Larger values reduce the range of values possible, including helping to bring the higher frequency coefficients, which are already typically small values, to zero. This shrinks the size of the file, as well as the amount of data that can be embedded.

The embedding rate directly affects the number of coefficients changed. For *SimpleSteg* the number is a strict percentage. *MBSteg* has a more complicated capacity measurement based on the number of times each value appears and an estimate of how many values can be modified while keeping an approximate shape of the global histogram.

Videos

A collection of standard YUV formatted videos were used as input to both algorithms. There is some variety in the videos in terms of content. Some are fairly static, like a video conference call or a news broadcaster, and others have a lot of motion, such as traveling down a highway, or panning the camera across a scene with moving objects. All of the videos were truncated to the first 150 frames to limit the time to process. They are all common raw YUV test videos. A list of the compressed sizes is presented in Table 1.

Because all of these videos are the same length, the difference in file sizes is

video	Q2 size	Q5 size
akiyo	161256	70683
bridge-close	594656	190128
bridge-far	332107	79374
carphone	478591	155334
claire	162909	63956
coastguard	923420	314816
container	221854	89186
foreman	608552	211669
grandma	254345	78485
hall	372085	122013
highway	617045	184077
miss-america	227756	57293
mobile	1736370	684755
mother-daughter	259454	87403
news	323714	141798
salesman	326987	118553
silent	340931	138884
suzie	358455	112802

Table 1. List of videos

due to a combination of two factors. One is the amount of movement between frames. The more movement there is, the more likely new blocks need to be coded, or blocks that have moved need larger changes. There are two types of movement. The simpler case is where objects in the frame are moving. Motion vectors can mostly cover this. The more complex case is camera motion. If the camera is panning, then new blocks are being added at the edges, and the rest of the blocks are moving. If the camera is zooming, then all of the blocks will change, although it depends on the exact scenery and the rate of zoom how much change this creates. The 'mobile' video has all of the above factors with the camera zooming out, panning left while viewing a toy train, which accounts for its much larger size.

Although embedding data into a video by slightly changing the DCT coefficients should not seem to change the size of the file, it does. This happens because

the coefficients are encoded using a variable length encoding scheme (VLC) where certain values might be much longer than others, even though they only differ by one. Because of the way that MBSteg models the DCT coefficients into buckets, the types of changes it makes are not least significant bit changes. That is, it will change an 1 to a 2 (or vice versa), which swaps two bits. This is different from our SimpleSteg which always only flips the last bit. The cumulative effect of these changes, when passed through the VLC encoding, is that MBSteg embedded files are slightly smaller (less than 1% up to 50% embedding) than the clean version, whereas SimpleSteg embedded files are slightly larger (less than 2% up to 50% embedding).

3.2 Video Steganalysis

To determine if a video is stegged using the above techniques, the first step will be to determine if the individual frames are stegged. Because of the similarity of JPEG and MPEG, and of the steganographic technique, it follows that the feature sets used in DCT-JPEG steganography will apply here as well. Once the frames are classified, another classification is done with those results to decide if the video as a whole is stegged.

As discussed in section 2.3, there are many feature sets available for images that could be used. Here only two are explored. The reasoning is that our prior work [5, 6] with the Fridrich/Pevný set shows its accuracy to be very high. The other feature set used is [7] which is a little bit more recent. While a good feature set is important, the methodology used here can easily be updated to newer feature sets as they become available.

3.2.1 Feature Sets Description

Through a series of papers[8, 9, 10, 11], Fridrich et al. developed a comprehensive set of features for DCT based steganalysis. The set used in the final paper can be broken up into seven parts:

- The first part looks at the global histogram of DCT values across the image. In particular, it is a count of the number of occurrences of the values $-5..5$ in any position in all blocks.
- The second part is a set of histograms for 5 of the lowest frequency AC coefficients, over the same $-5..5$ range.
- The third part are dual histograms across 9 of the lowest frequency AC coefficients, capturing distribution of the values.
- The fourth part is a measure variation across all the DCT modes.
- The fifth part is a measure of the "blockiness" of the image, measured in the spatial domain.
- The sixth part is a co-occurrence matrix of pairs of neighboring DCT coefficients.
- The seventh part uses a Markov process based approach that observes the difference of DCT modes across neighboring blocks.

The Liu feature set from [7] measures the joint occurrence of small valued DCT coefficients both within a block, and across adjacent blocks. For example, it will look at how many times AC coefficient (3,2) = 2 when AC coefficient (2,1) = 4, both within the same block, and then with both the block to the right and the block below. The range of coefficients to look for is a parameter, which the

authors set to $-6..6$ for a total of 169 features, and this value is also used in our testing.

3.2.2 Frame Classification

The Fridrich[11] feature set is used as a baseline. Liu's feature set, which is slightly more recent, gives a comparison point. Although extensions to these were planned that would included inter-frame statistics, the tests using next frame approximation, discussed next, indicated that this would be of limited value since the noise from motion outweighs the noise from the steganographic data.

For the approximation of the cover frame needed in the Fridrich set, three different approaches were taken.

cropping The first method is the same cropping and re-compression of the image done in [11]. This involves converting the frame back to the spatial domain, cropping the image by four pixels in both directions and then re-compressing the result. The reason for using 4 pixels is to cut the 8x8 block used in the DCT transform, but not significantly change the image. The re-compressed image will have similar DCT coefficients that can be used in the approximation.

next-frame The second approximation technique simply uses the next frame of the video, which should be fairly close to the current frame unless there is a scene cut, which happen infrequently.

frame-averaging The last method is one of the approximation methods from [12] that averaged frames on either side of the one under consideration. Averaging frames from either side can give two benefits. First, it reduces the effect from a scene change that the next-frame method might have. Second, frame under consideration could be considered the middle of the frames before and after,

so their average will be closer than either one of those frames. In [12] they find that in most cases one frame from each side is sufficient.

Note that to do either of these last two requires manually recalculating nearly all of the DCT values for most frames, since in most cases they do not appear in the bitstream. For example, following an I frame is either a P or B frame. Since these frame types usually do not encode much of the picture as full DCT values, instead being residual values from motion vectors, the full spatial frame needs to be re-transformed to DCT values. The cropping method also needs to do this transform for one the equivalent of one frame per frame, and so is the same as using the next frame in complexity.

3.2.3 Video Classification

The above describes only a decision process for an single frame. Since a video consists of several frames, and the goal is to decide if the video as a whole is stegged, a separate decision procedure is required. The obvious choice, taken in [12], may be a "majority rules" approach where if over half of the frames are reported as stegged, than the whole video is considered stegged. This requires decoding the entire video, which may be very costly.

As shown in section 4.2, a better solution might be use a sequential test where after each frame a decision can be made whether to accept it as stegged, reject it as non-stegged, or test the next frame. A statistical reference book [13] provides just such a method. If more stegged frames than would be predicted by the false positive rate appear, then the video is likely stegged. Conversely, if more clean frames than would be predicted by our false negative rate appear, then the video is likely not-stegged. Indeed, the method also allows for specifying how confident it should be in the determination.

A simplifying assumption made here is that either all frames have embedded

data, or no frames do. Some forms of relaxing that assumption have simple solutions, and others do not. For example, the assumption that data is embedding starts from the beginning of the video and continues until the entire message is encoded might be handled by looking for a fall-off in the number of stegged images over time. However, a steganographer that embeds randomly throughout a video and keeps the overall number of frames embedded below the frame classifier's false-positive rate would be much harder to detect.

List of References

[1] T. Pevný and J. Fridrich, "Detection of Double-Compression in JPEG Images for Applications in Steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, June 2008.

[2] D. Foti, "mpgwrite," Last retrieved June 2012, 1999. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/309

[3] D. Foti, "mpgread," Last retrieved June 2012, 1999. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/308-mpgread

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

[5] N. R. Bennett, "JPEG Steganalysis & TCP/IP Steganography," Ph.D. dissertation, University of Rhode Island, 2009.

[6] E. McCabe, "Development and Evaluation of Classification Tools for Steganalysis of JPEGs," Ph.D. dissertation, University of Rhode Island, 2008.

[7] Q. Liu, A. H. Sung, and M. Qiao, "Improved detection and evaluation for JPEG steganalysis," *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, p. 873, 2009.

[8] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Information Hiding*. Springer, 2003, pp. 310–323.

[9] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Information Hiding*. Springer, 2005, pp. 67–81.

[10] T. Pevný and J. Fridrich, "Towards multi-class blind steganalyzer for JPEG images," *Digital Watermarking*, pp. 39–53, 2005.

[11] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," *Proceedings of SPIE*, pp. 650 503–650 503–13, 2007.

[12] U. Budhia, "STEGANALYSIS OF VIDEO SEQUENCES USING COLLUSION SENSITIVITY," Ph.D. dissertation, Texas A&M University, 2005.

[13] E. Keeping, *Introduction to Statistical Inference*. Mineola, NY: Dover Publications, Inc., 1995.

CHAPTER 4

Evaluation

Evaluation is done into two parts. First, classification systems for each frame are considered based on the parameters discussed in section 3.2. Second, methods for classifying the entire video are analyzed.

4.1 Feature Selection

Evaluation started with checking the efficacy of different approximation methods for the cover image as well as varied sets of features. In particular, tests were performed on approximation by cropping, using the adjacent frame, and frame averaging. Both the features used in Fridrich and Pevný's paper [1] as well as newer features from Liu [2] were tested.

The original paper by Fridrich applied an SVM with a Gaussian kernel for classification. In our prior work with images[3], an LDA gave similar performance, with less overhead in model building. Early in this current project, a comparison of LDA and linear SVM's showed that LDA performance was superior, and so the research proceeded with that. For comparison purposes, another attempt to use radial kernels using *tune.svm* function, from R[4] package e1071[5], was performed. The *tune.svm* function will search the parameter space of the kernel function and return the best values for the given set of data. The recommended parameters were $C = 1, \gamma \approx 2^{-8.1}$. These parameters were then used against the entire data set. These new results do show better classification in most cases, particularly in some of the higher embedding rates. For example, at 20% embedding it correctly classifies Simple Steg and improves MBSteg by 7-8%. The low embedding rates of MBSteg embedded videos only improve by less than 2-6%. Preliminary results are shown in the Appendix.

To evaluate the detection technique, cross-validated, trained LDA models classified a test data set of videos that are stegged at different embedding rates. Two different steganographic techniques were tested. One of them is a modified version of MBSteg[6] that embeds data in each frame. The second is a more naïve implementation that just changes every AC DCT value with $abs(x) > 1$ up to the embedding rate percentage.

Previously, our JPEG work had very good accuracy down to 15% embedding [3]. Since one of the goals of steganalysis is to lower the effective bandwidth of a steganographic carrier, the effectiveness of the models was tested on several lower rates, as listed below.

Table 2 presents the parameters just outlined, adding just one more for quality of the video.

Parameter	Values
Feature Sets	Fridrich [1], Markov [2]
Cover Approximations	Crop, Frame averaging, Next
Steg Programs	SimpleSteg, MBSteg
Embedding Rates	5, 10, 15, 20, 50
Quality Levels	2, 5

Table 2. Parameters for evaluation

The cover approximations do not apply to the Markov process used in Liu's feature set. All other combinations of these parameters are valid and were tested. In the analysis below, note that expectation is that accuracy on videos with data embedded by MBSteg is lower than those from SimpleSteg, due to the differences in how the steganography programs work. Additionally, the accuracy is lower as the embedding rate goes down as there are fewer changes to detect.

Table 3 shows the error rates if a model is built over all frames of a video using only Fridrich's feature set with cropping approximation. The rates are much lower than expected based on the work with image classification. After looking at the

raw data, the problem is immediately obvious. The data from P and B frames is significantly different because they are not representing an entire picture. Because the features include statistics that compare adjacent blocks of the image, having missing data skews those statistics dramatically.

	Q-scale	Embed Rate	FP Rate	FN Rate	Overall Error
Simple Steg					
	2	5	35.3%	30.3%	32.8%
		10	28.0%	22.7%	25.4%
		15	21.4%	19.9%	20.7%
		20	18.3%	19.4%	18.8%
		50	12.7%	13.7%	13.2%
	5	5	52.6%	38.5%	45.6%
		10	47.2%	35.3%	41.2%
		15	45.2%	32.3%	38.8%
		20	39.8%	27.6%	33.7%
		50	20.4%	14.5%	17.5%
MBSteg					
	2	5	46.0%	34.8%	40.4%
		10	42.3%	26.2%	34.2%
		15	38.1%	22.1%	30.1%
		20	32.5%	22.9%	27.7%
		50	24.2%	21.5%	22.9%
	5	5	53.1%	41.9%	47.5%
		10	53.9%	37.4%	45.6%
		15	49.9%	34.1%	42.0%
		20	49.7%	33.1%	41.4%
		50	36.2%	25.5%	30.8%

Table 3. Error over all frames

Given that I, P and B frames are all going to have different statistics, it is logical to create different models for each type. Table 4 and Table 5 show the rates given by the classifier for P and B frames, again using Fridrich's features with cropping.

These error rates are also very high, again because of the missing data. Performing DCT-reconstruction, by re-compressing the spatial frame, would not increase the accuracy much since the effective embedding rate is so much lower in

	Q-scale	Embed Rate	FP Rate	FN Rate	Overall Error
Simple Steg					
	2	5	16.37%	19.67%	18.02%
		10	9.61%	7.51%	8.56%
		15	4.20%	3.75%	3.98%
		20	2.25%	2.55%	2.40%
		50	1.20%	0.75%	0.98%
	5	5	44.14%	31.53%	37.84%
		10	38.74%	26.13%	32.43%
		15	32.73%	20.42%	26.58%
		20	29.43%	18.32%	23.87%
		50	16.82%	19.07%	17.94%
MBSteg					
	2	5	46.70%	44.44%	45.57%
		10	37.39%	39.79%	38.59%
		15	35.44%	31.98%	33.71%
		20	30.93%	28.53%	29.73%
		50	7.21%	6.16%	6.68%
	5	5	56.01%	37.54%	46.77%
		10	51.35%	36.19%	43.77%
		15	47.30%	31.08%	39.19%
		20	45.50%	30.33%	37.91%
		50	33.33%	20.72%	27.03%

Table 4. Error over P frames

	Q-scale	Embed Rate	FP Rate	FN Rate	Overall Error
Simple Steg					
	2	5	42.73%	34.31%	38.52%
		10	36.03%	28.06%	32.04%
		15	29.78%	27.21%	28.49%
		20	26.06%	26.80%	26.43%
		50	18.61%	20.27%	19.44%
	5	5	48.28%	36.88%	42.58%
		10	47.31%	28.41%	37.86%
		15	44.27%	24.57%	34.42%
		20	37.51%	26.58%	32.04%
		50	29.95%	25.14%	27.55%
MBSteg					
	2	5	54.81%	36.88%	45.85%
		10	51.78%	34.59%	43.18%
		15	51.03%	31.27%	41.15%
		20	45.30%	27.95%	36.63%
		50	26.86%	18.61%	22.74%
	5	5	53.55%	42.15%	47.85%
		10	55.61%	38.20%	46.91%
		15	52.92%	35.45%	44.19%
		20	54.47%	34.71%	44.59%
		50	40.21%	28.92%	34.56%

Table 5. Error over B frames

P and B frames. Because of this, the steganalysis will proceed by looking only at the I frames of the video.

The next set of tables (Table 6 through Table 9) gather the results from tests run across only the I frames of each video.

It appears from Table 9 that the Liu feature set is not well suited to MPEG videos, as its best performance is a 12.5% error rate. That may be because of the low resolution of the videos. As their paper shows, accuracy of their method depends on image complexity, with lower complexity making detection easier. Because the video resolution is small, but generally has the same field of view as an image, its complexity is fairly high.

Using the Fridrich feature set, Tables 6 through 8 show that *Cropping* is the only approximation method to achieve 0% error on any group of videos. It also has consistently better performance by several percentage points on almost all sets, with the exception of some of the lower embedding rates, although all methods perform terribly there. The other two approximation methods have similar performance, with the average difference between them only being a couple of percentage points different in most cases. Only on SimpleSteg with a qscale of 2 does *Frame Averaging* do consistently better than *Next frame*.

In evaluating performance on individual videos, it is clear that videos with a lot of motion are classified more accurately by cropping than by the other approximation techniques. This indicates that the changes even in 1/30th of a second are too great to use adjacent frames as an reference, either independently or averaged with other nearby frames. From these results, it is clear that the best choices for future tests would be the features from [1] with the cropping method of cover approximation in all future tests. All the remaining data presented uses features extracted in this way.

Method	Quality	Embed %	FP Rate %	FN Rate %	Overall Error
Simple Steg					
	2	5	34%	30%	32%
		10	20%	26%	23%
		15	6%	11%	9%
		20	5%	11%	8%
		50	0%	0%	0%
	5	5	33%	29%	31%
		10	13%	9%	11%
		15	4%	8%	6%
		20	1%	7%	4%
		50	0%	0%	0%
MB Steg					
	2	5	53%	34%	44%
		10	41%	27%	34%
		15	29%	41%	35%
		20	22%	22%	22%
		50	9%	7%	8%
	5	5	41%	52%	47%
		10	48%	34%	41%
		15	33%	33%	33%
		20	24%	29%	26%
		50	12%	12%	13%

Table 6. Error rates using [1] under varying Quality and Embedding Rate using Cropping calibration

Method	Quality	Embed %	FP %	FN %	Overall Error
Simple Steg					
	2	5	42.13%	46.30%	44.21%
		10	34.26%	35.19%	34.72%
		15	27.78%	35.19%	31.48%
		20	23.61%	28.70%	26.16%
		50	3.70%	7.87%	5.79%
	5	5	35.19%	41.20%	38.19%
		10	30.56%	31.48%	31.02%
		15	22.22%	25.93%	24.07%
		20	18.52%	22.22%	20.37%
		50	7.41%	6.48%	6.94%
MBSteg					
	2	5	40.28%	46.30%	43.29%
		10	30.09%	43.06%	36.57%
		15	27.78%	36.11%	31.94%
		20	29.63%	28.24%	28.94%
		50	12.50%	17.13%	14.81%
	5	5	43.06%	46.76%	44.91%
		10	35.65%	39.81%	37.73%
		15	36.11%	35.65%	35.88%
		20	29.17%	30.56%	29.86%
		50	14.35%	18.06%	16.20%

Table 7. Error rates using [1] under varying Quality and Embedding Rate using Next Frame calibration

Method	Quality	Embed %	FP %	FN %	Overall Error
Simple Steg					
	2	5	48.00%	32.00%	40.00%
		10	26.00%	28.00%	27.00%
		15	19.00%	30.00%	25.00%
		20	13.00%	21.00%	17.00%
		50	3.00%	7.00%	5.00%
	5	5	42.00%	39.00%	41.00%
		10	37.00%	32.00%	34.00%
		15	26.00%	23.00%	25.00%
		20	17.00%	25.00%	21.00%
		50	6.00%	7.00%	7.00%
MBSteg					
	2	5	44.00%	43.00%	43.00%
		10	38.00%	35.00%	37.00%
		15	33.00%	37.00%	35.00%
		20	28.00%	33.00%	30.00%
		50	11.00%	9.00%	10.00%
	5	5	48.00%	45.00%	47.00%
		10	37.00%	39.00%	38.00%
		15	38.00%	34.00%	36.00%
		20	35.00%	29.00%	32.00%
		50	14.00%	14.00%	14.00%

Table 8. Error rates using [1] under varying Quality and Embedding Rate using Frame Averaging calibration

Method	Quality	Embed %	FP %	FN %	Overall Error
Simple Steg					
	2	5	32.87%	20.83%	26.85%
		10	33.33%	15.74%	24.54%
		15	23.61%	20.83%	22.22%
		20	23.61%	17.13%	20.37%
		50	21.76%	10.19%	15.97%
	5	5	45.37%	37.50%	41.44%
		10	51.39%	25.93%	38.66%
		15	50.00%	22.69%	36.34%
		20	50.00%	24.07%	37.04%
		50	50.93%	18.06%	34.49%
MBSteg					
	2	5	38.89%	45.83%	42.36%
		10	32.87%	39.81%	36.34%
		15	27.78%	33.80%	30.79%
		20	22.69%	29.17%	25.93%
		50	12.04%	12.96%	12.50%
	5	5	30.09%	65.28%	47.69%
		10	24.54%	59.72%	42.13%
		15	23.15%	58.80%	40.97%
		20	23.15%	56.94%	40.05%
		50	15.28%	35.65%	25.46%

Table 9. Error rates using [2] under varying Quality and Embedding Rate

4.2 Video Classification

The above results give error rates on a frame by frame basis. The goal now is to classify the video as a whole. In [7] they use a simple majority rule, so if more frames are classified as stegged, then the video is considered stegged. The effect of varying that threshold is shown in Table 10. The table shows the error rates for each quality level over all embedding rates.

Threshold	SS/Q2	SS/Q5	MB/Q2	MB/Q5
0.1	0.16	0.14	0.33	0.34
0.2	0.16	0.11	0.28	0.33
0.3	0.11	0.08	0.27	0.32
0.4	0.09	0.09	0.24	0.32
0.45	0.11	0.07	0.23	0.27
0.5	0.08	0.06	0.19	0.26
0.55	0.1	0.06	0.22	0.29
0.6	0.09	0.07	0.25	0.29
0.7	0.11	0.07	0.24	0.29
0.8	0.16	0.08	0.29	0.31
0.9	0.18	0.11	0.3	0.32

Table 10. Video error rates at various frame thresholds

Because the false positive and false negative rates are similar (see Table 6), a majority rule does, in fact, give the lowest overall error. That is, from the table it is evident that using a threshold of 50% of the frames has the lowest error rate. Note that this is also due to similar prevalence in the test set. In real-world use, this might need adjustment. In the test data, there are 50% stegged videos and 50% clean videos. A real data set will have a much higher number of clean videos, so the classifier might be adjusted to expect a lower prevalence of steg. However making this adjustment also make it less reliable for positively identifying stegged video frames. See [8] for a full description of how adjusting the LDA cutoff changes the Positive Predictive Value for the image classifier.

Table 11 shows the results of using the majority rule. These results are better

	Q-Scale	Embed Rate	FP Rate	FN Rate	Overall Error
Simple Steg					
	2	5	22.0%	22.0%	22.2%
	2	10	11.0%	22.0%	16.7%
	2	15	0.0%	0.0%	0.0%
	2	20	0.0%	6.0%	2.8%
	2	50	0.0%	0.0%	0.0%
	5	5	28.0%	22.0%	25.0%
	5	10	0.0%	6.0%	2.8%
	5	15	0.0%	6.0%	2.8%
	5	20	0.0%	0.0%	0.0%
	5	50	0.0%	0.0%	0.0%
MBSteg					
	2	5	56.0%	22.0%	38.9%
	2	10	22.0%	17.0%	19.4%
	2	15	17.0%	33.0%	25.0%
	2	20	17.0%	6.0%	11.1%
	2	50	6.0%	0.0%	2.8%
	5	5	33.0%	44.0%	38.9%
	5	10	50.0%	22.0%	36.1%
	5	15	28.0%	28.0%	27.8%
	5	20	11.0%	22.0%	16.7%
	5	50	11.0%	6.0%	8.3%

Table 11. Video classification errors with majority rule

than those for the individual frames as shown in Table 6. This is because the error rates of the frame classifier are low enough that the percentage of clean or stegged frames can stabilize above the threshold.

There is one further optimization that could be made when classifying video. Because decoding video is an expensive operation, it is desirable to stop processing as soon as decision can be reached. What is necessary is a method that looks at the cumulative results as it processes each frame and decides whether or not it has seen sufficient evidence to classify the entire video. To do this, a method found in [9] for a sequential test of a binomial distribution seems appropriate.

The method uses four parameters, α, β, θ_0, and θ_1. It then requires tracking the number of frames marked as non-stegged, d_m, and the number of frames processed, m. The parameters α and β are used to control confidence bounds on overall classification. False positives are controlled by α and false negative errors by β. To obtain 95% confidence, both are set to 0.025. The θ_0 and θ_1 parameters control the proportion of stegged or non-stegged frames below which the mistakes may be from the frame classifier.

From these values, the method computes A_m, the accepting boundary, and R_m, for rejection, as shown below in Equation 1 and Equation 2. Equation 3 shows the constraint under which it must continue processing the video.

$$A_m = \frac{\log \frac{\beta}{1-\alpha} + m \log \frac{1-\theta_0}{1-\theta_1}}{\log \frac{\theta_0}{\theta_1} + \log \frac{1-\theta_0}{1-\theta_1}} \tag{1}$$

$$R_m = \frac{\log \frac{1-\beta}{\alpha} + m \log \frac{1-\theta_0}{1-\theta_1}}{\log \frac{\theta_0}{\theta_1} + \log \frac{1-\theta_0}{1-\theta_1}} \tag{2}$$

$$A_m < d_m < R_m \tag{3}$$

If the false positive rate of α_f and false negative rate of β_f for the frame classifier, then $\theta_0 = \beta_f$ and $\theta_1 = 1 - \alpha_f$.

If $d_m < A_m$, then the video is considered stegged, because of the m frames processed, more of them are classified stegged than would be expected by the false positive rate and a margin wide enough for α. On the other hand, if $d_m > R_m$, then more frames are classified as non-stegged than expected for the frame classifier's false negative rate plus the margin of β, so the video must not be stegged.

Two plots demonstrating this method are shown in Figure 3. The upper figure shows a non-stegged video tested with the parameters for a SimpleSteg 10% embedded video of with a qscale of 2. The second shows a stegged video with an embedding rate of 20% and a qscale of 2. Note that the distance between the accepting and rejecting lines is further apart at the lower embedding rate because there is more uncertainty in the underlying frame classification.

Table 12 shows the result of the sequential test. Table 13 shows the comparison between the sequential test and the majority test. The sequential test is generally lower than that of the majority rule, especially for the low embedding rates. They both perform equally well, perfectly in fact, in the easiest case of 50% embedding using SimpleSteg. They are also comparable for MBSteg at that embedding rate, with majority rule getting 2.8% error, and Sequential getting 3% error for qscale=2. At the lower embedding rates it seems the accuracy is much worse, with 50% for the sequential test while at 20-25% for the majority rule at the 10-15% embedding rate.

The reason the sequential test does worse is because the underlying classifiers have high false positive and false negative error rates, it makes it difficult for the sequential test to complete and be confident in the result. The videos are very short clips, so it is possible that a longer test might yield better results. For the higher embedding rates, however, the sequential tests are nearly as accurate and only need to process a fraction of the frames to make that determination.

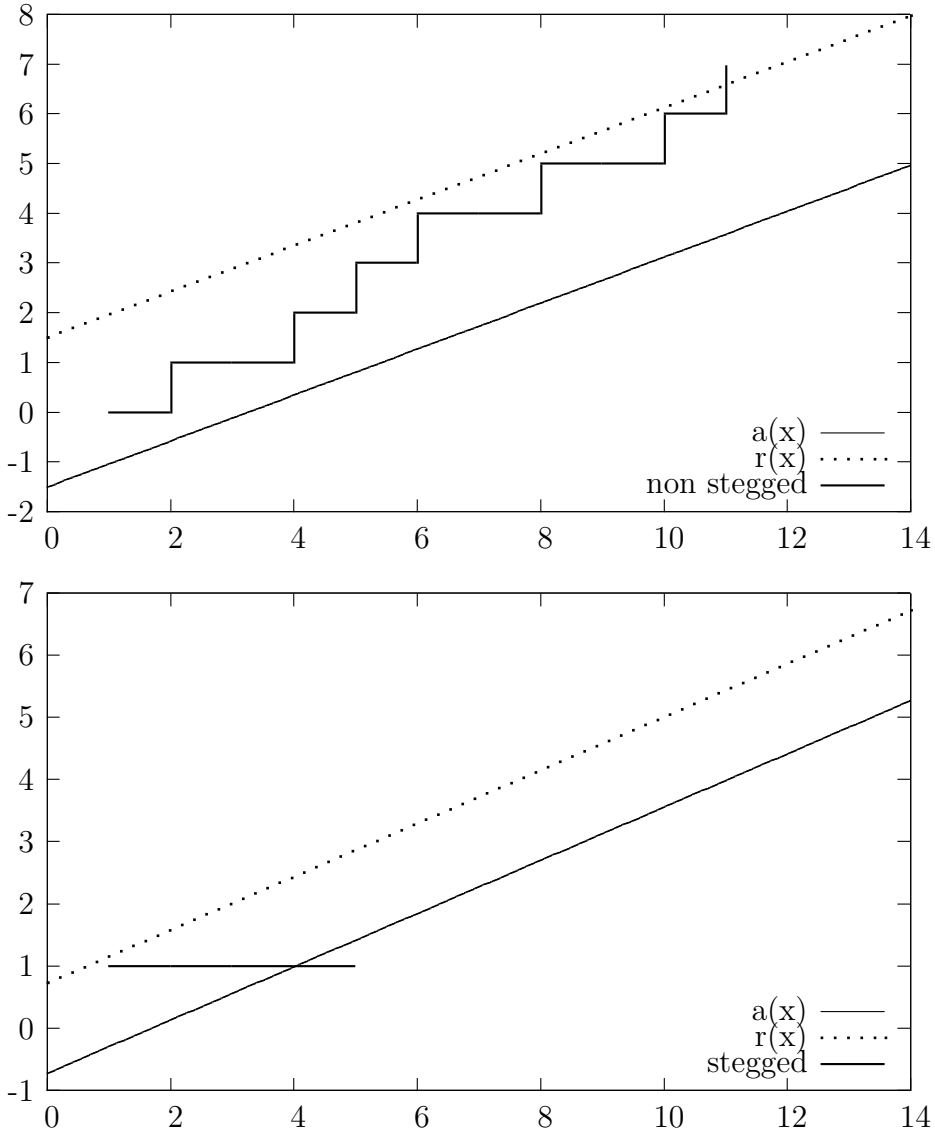

Figure 3. Decision process for non-stegged (top) and stegged (bottom) videos

	Q-scale	Embed Rate	θ_0	θ_1	FP Rate	FN Rate	Undecided	Overall Accuracy	% Frames processed
Simple Steg									
	2	5	0.34	0.7	5.56%	11.11%	36.1%	56.0%	75%
	2	10	0.2	0.74	0.00%	16.67%	11.1%	81.0%	44%
	2	15	0.06	0.89	0.00%	0.00%	0.0%	100.0%	19%
	2	20	0.05	0.89	0.00%	11.11%	0.0%	94.0%	18%
	2	50	0.05	0.95	0.00%	0.00%	0.0%	100.0%	17%
	5	5	0.33	0.71	27.78%	11.11%	16.7%	64.0%	61%
	5	10	0.13	0.91	5.56%	5.56%	2.8%	92.0%	22%
	5	15	0.04	0.92	0.00%	5.56%	0.0%	97.0%	18%
	5	20	0.01	0.93	5.56%	0.00%	2.8%	97.0%	13%
	5	50	0.05	0.95	0.00%	0.00%	0.0%	100.0%	17%
MBSteg									
	2	5	0.53	0.66	11.11%	0.00%	86.1%	8.0%	97%
	2	10	0.41	0.73	22.22%	11.11%	33.3%	50.0%	73%
	2	15	0.29	0.59	11.11%	22.22%	33.3%	50.0%	69%
	2	20	0.22	0.78	16.67%	11.11%	2.8%	83.0%	34%
	2	50	0.09	0.93	5.56%	0.00%	0.0%	97.0%	20%
	5	5	0.41	0.48	0.00%	0.00%	100.0%	0.0%	97%
	5	10	0.48	0.66	16.67%	11.11%	58.3%	28.0%	92%
	5	15	0.33	0.67	11.11%	22.22%	33.3%	50.0%	71%
	5	20	0.24	0.71	5.56%	16.67%	11.1%	78.0%	53%
	5	50	0.12	0.88	11.11%	11.11%	0.0%	89.0%	19%

Table 12. Sequential test parameters and accuracy

	Q-Scale	Embed Rate	Threshold	Sequential
Simple Steg				
	2	5	77.8%	56.0%
	2	10	88.3%	81.0%
	2	15	100.00%	100.0%
	2	20	97.2%	94.0%
	2	50	100.0%	100.0%
	5	5	75.0%	64.0%
	5	10	97.2%	92.0%
	5	15	97.2%	97.0%
	5	20	100.0%	97.0%
	5	50	100.0%	100.0%
MBSteg				
	2	5	61.1%	8.0%
	2	10	88.6%	50.0%
	2	15	75.0%	50.0%
	2	20	88.9%	83.0%
	2	50	97.2%	97.0%
	5	5	61.1%	0.0%
	5	10	63.9%	28.0%
	5	15	72.2%	50.0%
	5	20	83.3%	78.0%
	5	50	91.7%	89.0%

Table 13. Comparison Video classification accuracy with threshold vs. sequential

List of References

[1] T. Pevný and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," *Proceedings of SPIE*, pp. 650 503–650 503–13, 2007.

[2] Q. Liu, A. H. Sung, and M. Qiao, "Improved detection and evaluation for JPEG steganalysis," *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, p. 873, 2009.

[3] N. R. Bennett, "JPEG Steganalysis & TCP/IP Steganography," Ph.D. dissertation, University of Rhode Island, 2009.

[4] "R." [Online]. Available: http://www.r-project.org/

[5] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel, "Misc functions of the department of statistics (e1071), tu wien," *R package*, pp. 1–5, 2008.

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

[7] U. Budhia, "STEGANALYSIS OF VIDEO SEQUENCES USING COLLUSION SENSITIVITY," Ph.D. dissertation, Texas A&M University, 2005.

[8] E. McCabe, "Development and Evaluation of Classification Tools for Steganalysis of JPEGs," Ph.D. dissertation, University of Rhode Island, 2008.

[9] E. Keeping, *Introduction to Statistical Inference*. Mineola, NY: Dover Publications, Inc., 1995.

CHAPTER 5

Conclusion

To summarize the above process, this research has determined that by using the best available steganography detection for DCT-embedded JPEG images, applying it to only the I-frames of an MPEG, a fairly accurate video steganalysis can be created. Furthermore, for medium and high embedding rates, a statistical test can be performed during processing that will minimize the time taken to classify the video. This is important because of the proliferation of video content.

When compared with the results from [1], the results above show that detecting steganographic data in MPEG is significantly more difficult than in JPEG data. Those results achieved, even at 5% embedding rates, an overall accuracy of 93-95%. The reason may partly be because of the quantization done in MPEG videos. Although both formats use similar techniques, the quantization matrices from MPEG are not derived in the same way as JPEG. The MPEG specification was designed for small images and low bandwidth. This means that quality of the image is generally very poor. Even in the image results, there is a drop off in classification as the quality of the image decreases. This happens because more of the data is forced to zero, so no useful statistics can be collected.

Detecting steganography is only the first step for investigators. The next step is to extract the data. While some work in this has been done, such as in stegbreak [2], it is necessarily incomplete. The first issue is deciding which of several steganographic techniques was applied. Second it is necessary to decide which particular program embedded the data (see [3] for an example of multi-class analysis), and in some cases even the version is important. Third, in order to extract the data most steganographic programs require a password. If the

program uses any sort of encryption, and most do, then testing each password requires expensive calculations.

Because of this, even relatively low false positive rates means spending an inordinate amount of time trying to crack encryption, only to find no data is there. Deciding that the correct data has been extracted is also a challenge. Some steganographic programs will tell the user if data was extracted successfully, however from a steganographer's point of view that is a bad design. Stegbreak measures the entropy of the extracted data, which works well if the embedded data is text or a well known file format. However, if a steganographer is aware of this technique, they could easily encrypt their data before embedding it, where it will be encrypted again. This would easily foil the entropy test, as encrypted data should be indistinguishable from noise.

In the statistical analysis of the image classifier provided in [4], it is shown that with some manipulation of the threshold used, the classifier can be more selective in deciding to classify an image or frame as stegged. It also shows that these classifiers work best when the prevalence of steganography nearly one-to-one with non-stegged media. This is obviously not the case. Therefore an examiner must work to pre-classify data by other means, such as being part of other correspondence, or by time line information that correlates with the investigation.

While steg-analysts continue to make headway in the arms race with steganographers, a good steganographer will always have the upper hand. The variety of techniques and places to hide data are nearly unlimited, and steganalysis cannot begin without some knowledge of the technique employed. Even given the caveats listed here, hopefully this research will deter some steganographers from assuming video is a high-capacity way to get their message out.

Although this work focused on MPEG videos, almost all other video codecs use

DCT and quantization to reduce the amount of data and make further compression possible. Other codecs also have similar concepts of key and differential frames, but differ in the spacing, calculation, and encoding. To use the steganalysis performed in this project in other codecs, some analysis of the spread of coefficient values through the frame types to see if the assumptions of the feature set used still hold. Different models might need to be built for various frame types. Most of the newer codecs have greater distance between key frames, so models specific to differential frames will become more important. Some newer codecs allow the quantizer to change within a frame, to adjust the level of detail in part of the image. This might mean re-evaluating the feature sets which currently build models that are quality level dependent. Additionally, other frame classifiers might yield more accurate results, as indicated by preliminary testing of the Gaussian kernel SVM shown in the Appendix.

List of References

[1] N. R. Bennett, "JPEG Steganalysis & TCP/IP Steganography," Ph.D. dissertation, University of Rhode Island, 2009.

[2] N. Provos, "Stegbreak," Last retrieved June 2012, 2004. [Online]. Available: http://www.outguess.org/detection.php

[3] T. Pevný and J. Fridrich, "Towards multi-class blind steganalyzer for JPEG images," *Digital Watermarking*, pp. 39–53, 2005.

[4] E. McCabe, "Development and Evaluation of Classification Tools for Steganalysis of JPEGs," Ph.D. dissertation, University of Rhode Island, 2008.

APPENDIX

Preliminary SVM Results

As explained in Chapter 4, a late development included testing an SVM model for frame classification instead of the LDA model. The results here show the error rates of a radial kernel using the parameters $C = 1, \gamma = 2^{-8}$.

	Q-scale	Embed Rate	FP Rate	FN Rate	Overall Error
Simple Steg					
	2	5	12.5%	14.8%	13.7%
		10	4.2%	4.2%	4.2%
		15	1.9%	0.5%	1.2%
		20	0.0%	0.0%	0.0%
		50	0.0%	0.0%	0.0%
	5	5	10.2%	13.4%	11.8%
		10	0.9%	4.6%	2.8%
		15	0.5%	0.9%	0.7%
		20	0.0%	0.0%	0.0%
		50	2.3%	0.0%	1.2%
MBSteg					
	2	5	46.3%	34.3%	40.3%
		10	38.0%	21.8%	29.9%
		15	29.2%	10.6%	19.9%
		20	17.6%	10.2%	13.9%
		50	2.3%	1.9%	2.1%
	5	5	77.3%	11.6%	44.4%
		10	47.2%	23.1%	35.2%
		15	39.8%	12.0%	25.9%
		20	27.3%	11.1%	19.2%
		50	6.5%	0.0%	3.2%

BIBLIOGRAPHY

"Matlab." [Online]. Available: http://mathworks.com/products/matlab/

"R." [Online]. Available: http://www.r-project.org/

Avcibaş, I., Memon, N., and Sankur, B., "Steganalysis using image quality metrics." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 12, no. 2, pp. 221–9, Jan. 2003.

Bennett, N. R., "JPEG Steganalysis & TCP/IP Steganography," Ph.D. dissertation, University of Rhode Island, 2009.

Budhia, U., "STEGANALYSIS OF VIDEO SEQUENCES USING COLLUSION SENSITIVITY," Ph.D. dissertation, Texas A&M University, 2005.

Budhia, U. and Kundur, D., "Digital video steganalysis exploiting collusion sensitivity," *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III*, vol. 5403, pp. 210—-221, 2004.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., and Weingessel, A., "Misc functions of the department of statistics (e1071), tu wien," *R package*, pp. 1–5, 2008.

Eskicioglu, A. and Elbasi, E., "Robust DWT Based MPEG-1 Watermarking in Four Bands," *The Second Secure Knowledge Management Workshop (SKM), New York City, NY*, 2006.

Farid, H. and Lyu, S., "Higher-order Wavelet Statistics and their Application to Digital Forensics 2 . Image Statistics," *IEEE Workshop on Statistical Analysis in Computer Vision*, 2003.

Foti, D., "mpgread," Last retrieved June 2012, 1999. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/308-mpgread

Foti, D., "mpgwrite," Last retrieved June 2012, 1999. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/309

Fridrich, J., "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Information Hiding*. Springer, 2005, pp. 67–81.

Fridrich, J., Goljan, M., and Du, R., "Reliable detection of LSB steganography in color and grayscale images," *Proceedings of the 2001 workshop on Multimedia and security new challenges - MM&Sec '01*, p. 27, 2001.

Fridrich, J., Goljan, M., and Hogea, D., "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Information Hiding*. Springer, 2003, pp. 310–323.

Fu, D., Shi, Y., Zou, D., and Xuan, G., "JPEG Steganalysis Using Empirical Transition Matrix in Block DCT Domain," *2006 IEEE Workshop on Multimedia Signal Processing*, pp. 310–313, Oct. 2006.

Goljan, M., Fridrich, J., and Du, R., "Distortion-free data embedding for images," in *Information Hiding*. Springer, 2001, pp. 27–41.

Gupta, A. and Gupta, P., "Digital Watermarking of MPEG-4 Videos," Indian Institute of Technology Kanpur, Tech. Rep., 2003.

Hartung, F. and Girod, B., "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, May 1998.

Hetzl, S. and Mutzel, P., "A graph–theoretic approach to steganography," in *Communications and Multimedia Security*. Springer, 2005, pp. 119–128.

Hsu, C.-T. and Wu, J.-L., "DCT-Based Watermarking for Video," *IEEE Transaction on Consumer Electronics*, vol. 44, no. 1, pp. 206–216, Jan. 1998.

ISO 11172-2:2003, *ISO 11172, Part 2: Video*. ISO, Geneve, Switzerland, 2003.

Jainsky, J. S., Kundur, D., and Halverson, D. R., "Towards digital video steganalysis using asymptotic memoryless detection," *Proceedings of the 9th workshop on Multimedia & security - MM&Sec '07*, p. 161, 2007.

Keeping, E., *Introduction to Statistical Inference*. Mineola, NY: Dover Publications, Inc., 1995.

Langelaar, G. C. and Lagendijk, R. L., "Optimal differential energy watermarking of DCT encoded images and video." *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 10, no. 1, pp. 148–58, Jan. 2001.

Liu, B., Liu, F., and Wang, P., "Inter-frame Correlation Based Compressed Video Steganalysis," *2008 Congress on Image and Signal Processing*, pp. 42–46, 2008.

Liu, Q., Sung, A. H., and Qiao, M., "Improved detection and evaluation for JPEG steganalysis," *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, p. 873, 2009.

Lyu, S. and Farid, H., "Steganalysis Using Higher-Order Image Statistics," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

Lyu, S., Rockmore, D., and Farid, H., "A digital technique for art authentication." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 49, pp. 17 006–10, Dec. 2004.

Marvel, L., Boncelet Jr, C., and Retter, C., "Spread spectrum image steganography," *Image Processing, IEEE Transactions on*, vol. 8, no. 8, pp. 1075–1083, 2002.

McCabe, E., "Development and Evaluation of Classification Tools for Steganalysis of JPEGs," Ph.D. dissertation, University of Rhode Island, 2008.

Mobasseri, B., "Direct sequence watermarking of digital video using m-frames," *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, pp. 399–403, 1998.

Mobasseri, B. G. and Marcinak, M. P., "Watermarking of MPEG-2 video in compressed domain using VLC mapping," *Proceedings of the 7th workshop on Multimedia and security - MM&Sec '05*, pp. 91—-94, 2005.

Noda, H., Furuta, T., Niimi, M., and Kawaguchi, E., "Application of BPCS Steganography to Wavelet Compressed Video," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, no. 1, 2004, pp. 2147–2150.

Pevný, T. and Fridrich, J., "Towards multi-class blind steganalyzer for JPEG images," *Digital Watermarking*, pp. 39–53, 2005.

Pevný, T. and Fridrich, J., "Detection of Double-Compression in JPEG Images for Applications in Steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, June 2008.

Pevný, T. and Fridrich, J., "Merging Markov and DCT features for multi-class JPEG steganalysis," *Proceedings of SPIE*, pp. 650 503–650 503–13, 2007.

Provos, N., "Stegbreak," Last retrieved June 2012, 2004. [Online]. Available: http://www.outguess.org/detection.php

Provos, N., "Defending against statistical steganalysis," in *10th USENIX Security Symposium*, vol. 10. Citeseer, 2001, pp. 323–336.

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

Sarkar, A., Madhow, U., Chandrasekaran, S., and Manjunath, B. S., "Adaptive MPEG-2 video data hiding scheme," *Proceedings of SPIE*, pp. 65 051D–65 051D–9, 2007.

Shi, Y., Chen, C., and Chen, W., "A Markov process based approach to effective attacking JPEG steganography," in *Information Hiding*. Springer, 2007, pp. 249–264.

Simitopoulos, D., Tsaftaris, S. a., Boulgouris, N. V., Briassouli, A., and Strintzis, M. G., "Fast Watermarking of MPEG-1/2 Streams Using Compressed-Domain Perceptual Embedding and a Generalized Correlator Detector," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 8, pp. 1088–1106, 2004.

Solanki, K., Sarkar, A., and Manjunath, B., "YASS: Yet another steganographic scheme that resists blind steganalysis," in *Proceedings of the 9th international conference on Information hiding*. Springer-Verlag, 2007, pp. 16–31.

Su, K., Kundur, D., and Hatzinakos, D., "Spatially localized image-dependent watermarking for statistical invisibility and collusion resistance," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 52–66, Feb. 2005.

Vatolin, D. and Petrov, O., "MSU StegoVideo," 2007.

Xu, C., "Steganography in Compressed Video Stream," *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, pp. 269–272, 2006.

Xu, C. and Ping, X., "A Steganographic Algorithm in Uncompressed Video Sequence Based on Difference between Adjacent Frames," *Fourth International Conference on Image and Graphics (ICIG 2007)*, pp. 297–302, Aug. 2007.