STEGANALYZING TEXTURE IMAGES

Chunhua Chen[1], Yun Q. Shi[1], and Guorong Xuan[2]

[1] New Jersey Institute of Technology, Newark, NJ, USA ({cc86, shi}@njit.edu)
[2] Tongji University, Shanghai, China

ABSTRACT

A texture image is of noisy nature in its spatial representation. As a result, the data hidden in texture images, in particular in raw texture images, are hard to detect with current steganalytic methods. We propose an effective universal steganalyzer in this paper, which combines features, i.e., statistical moments of 1-D and 2-D characteristic functions extracted from the spatial representation and the block discrete cosine transform (BDCT) representations (with a set of different block sizes) of a given test image. This novel scheme can greatly improve the capability of attacking steganographic methods applied to texture images. In addition, it is shown that this scheme can be used as an effective universal steganalyzer for both texture and non-texture images.

Index Terms—steganalysis, texture images, block discrete cosine transform, rake transform

1. INTRODUCTION

Image steganalysis is referred to as the art and science to detect hidden message in images for covert communications. Several universal steganalyzers have been developed since the beginning of this century. Farid proposed a universal steganalysis method in [1], which can achieve detection accuracy generally better than random guess. This method uses quadrature mirror filters to decompose a test image into wavelet subbands first. The higher order statistics are thereafter calculated from wavelet coefficients and their prediction-errors of each high-frequency subband to form features. In [2], another universal steganalysis system was proposed by Shi et al. Different from [1], the statistical moments of the characteristic function (CF) (instead of the histogram) of a given image, its prediction-error image, and their discrete wavelet transform (DWT) subbands (including all of the low-low subbands), are selected as features. Compared to [1], this method improves steganalysis performance noticeably.

Texture images are of noisy nature. Hawkins describes image texture in detail in [3]: "The notion of texture appears to depend upon three ingredients: (1) some local 'order' is repeated over a region which is large in comparison to the order's size, (2) the order consists in the nonrandom arrangement of elementary parts, and (3) the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region." When steganographic tools are applied to a texture image and in particular when the resultant stego images are raw images, the embedded signal is submerged, which causes difficulty in steganalysis. This was observed during our research on steganalysis in 2004 [4]. Specifically, while our developed steganalytic method [2] performs quite well on 1096 natural images in the CorelDraw Version 10.0 software CD #3 [5], its performance deteriorates dramatically in steganalyzing texture images, especially when the resultant stego images are raw images. The effect of texture images on steganalysis was also reported by Bohme [6]. There it is found that images with noisy textures yield least accurate detection results for two steganalytic methods, i.e., *Regular-Singular* (RS) analysis and *Weighted Stego Image* (WS) analysis proposed by Fridrich, and conversely, both of these two steganalytic methods work well on images with flat regions and soft gradients. Thus with the least detectability, texture images may be selected as the cover for steganography. Effective steganalyzers specifically designed for texture images are therefore called for.

In this paper, we propose a novel steganalysis scheme, which combines features (statistical moments of 1-D and 2-D characteristic functions) extracted from the spatial representation and the block discrete cosine transform (BDCT) representations (with a set of different block sizes) of a given test image. With this newly developed steganalyzer, the steganalysis capability for texture images is greatly improved. In addition, this steganalyzer can be used as an effective universal steganalyzer for both texture and generally smooth images.

The rest of this paper is organized as follows. In Section 2, we describe the proposed steganalyzer. Experiments and results are presented in Section 3. Finally, a summary is given in Section 4.

2. PROPOSED STEGANALYZER

As mentioned in Section 1, the method [2] works well on the CorelDraw images but its performance on texture images is not satisfactory, as shown in Tables 2 (Section 3). The method [2] is partially based on the first three order statistical moments of the characteristic function, which is defined as the discrete Fourier transform (DFT) of the histogram of a given image. It has proven that after a message is embedded into an image, the above-defined moments will decrease or remain the same under the assumption that the hidden data is additive to and independent of the cover image and in such a distribution manner that the magnitude of its CF is non-increasing from 1 to N/2 [2]. Statistical investigation has shown that, the above-defined moments derived from the CorelDraw images decrease more than that derived from the 798 texture images used in our reported experimental work in this paper after the same data embedding process (e.g., generic Least Significant Biplane (LSB) replacement embedding [7] at 0.3 bpp (bits per pixel)). This observation partially explains the reason why the method [2] works effectively on CorelDraw images but its performance on texture images is unsatisfactory. Therefore, it is necessary to introduce new features to steganalyze texture images.

In [8], Chen et al proposed a universal steganalysis scheme, which performs well in attacking modern JPEG (Joint Photographic Experts Group) steganographic tools [9, 10, and 11]. This scheme utilizes features extracted from an image's JPEG representation, where the message is embedded, in addition to features from the image's spatial representation. This inspires us that combining features derived from the domain in which the data embedding is operated and features derived from a transform domain may enhance the steganalysis capability.

In this newly proposed steganalysis scheme, in addition to features generated from the spatial representation of a given image, we propose to add features generated from the block discrete cosine transform (BDCT) representations with a set of different block sizes. Referred to as "rake transform", these BDCT's can greatly improve the steganalysis performance. The feature generation procedure also involves prediction-error, DWT, 1-D (one-dimensional) and 2-D characteristic functions, and moments/marginal moments. Shown in Figure 1 (a) is the block diagram of our proposed feature generation scheme. The moment extraction block in Figure 1 (a) is given in detail in Figure 1 (b).

Similar to a theoretical analysis in [2], we can show that, the moments of CF of the BDCT coefficient 2-D arrays will not increase after data embedding, under the assumption that the transformed hidden data are additive to and independent of the cover image, and in such a distribution manner that the magnitude of its characteristic function is non-increasing from 1 to N/2 (where N is the total number of value levels of BDCT coefficients).

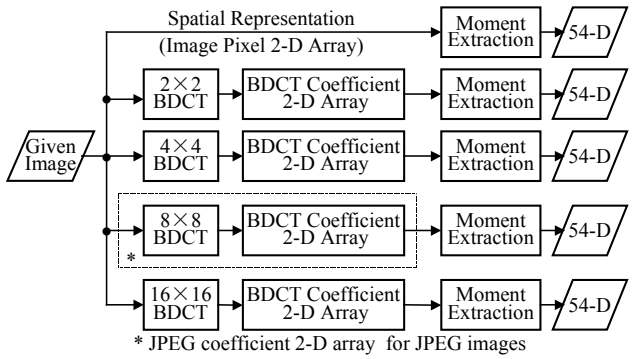
2.1. Rake transform

The block discrete cosine transform has been widely used in image/video compression applications due to its efficiency on decorrelation and energy compaction. For example, 8×8 BDCT has been adopted in JPEG and MPEG-2 (Moving Picture Experts Group) standards.

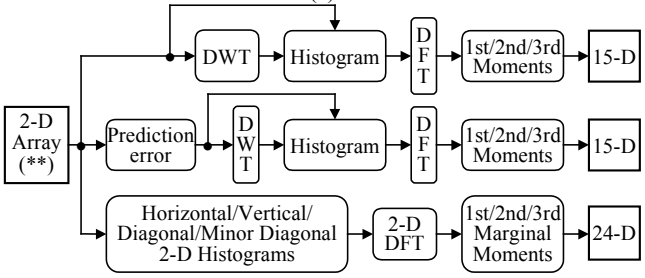
We propose to use block discrete cosine transform with a set of different block sizes in this novel steganalysis scheme. This is to utilize the comprehensive decorrelation capability of BDCT with various block sizes. Texture images are of regularity in pattern arrangement and therefore of regularity in local frequency distribution as well. Data embedding procedure changes the local frequency distribution of the texture images. With various block sizes, it is expected that this complicated frequency change can be perceived in BDCT coefficients with different block sizes and hence the data embedded in texture images may be detected with features extracted from these BDCT's. This expectation has been verified in our experimental results.

The procedure of N×N BDCT is described as follows. Firstly, the given image is divided into non-overlapping N×N blocks. Then, 2-D discrete cosine transform (DCT) is applied to each block independently. Finally, we obtain a 2-D array consisting of BDCT coefficients of all these blocks.

With each block size, we can obtain one 2-D array, from which we can generate one subset of features. As shown in Subsection 3.3, each feature subset associated with a specific block size contributes to the steganalyzer, just like each reflected signal contributes to the rake receiver to improve the SNR (signal to noise ratio) of the received signal, which is widely used in wireless communication applications. Therefore, we refer to these BDCT's with a set of block sizes as rake transform.



(a)



** Image pixel 2-D array/BDCT coefficient 2-D array/JPEG coefficient 2-D array

(b)

Figure 1. Block diagram of feature generation: (a) block diagram, (b) sub-block diagram for moment extraction

In this implementation, we choose the set of block sizes as 2×2, 4×4, 8×8, and 16×16 because this choice is of computational benefits in implementing DCT. Our experimental investigation also shows that this implementation gives similar performance to that of the implementation with a set of block sizes from 2×2, 3×3, …, to 16×16. In fact, without 3×3, 5×5, …, 15×15, performance change of the steganalyzer is not substantial. Moreover, when we include block size 32×32, the performance of the steganalyzer does not improve much but the computational costs rise, which include those of feature calculation, training, and testing. This is expected because the correlation between image pixels has become rather weak as the distance between pixels is too large.

Especially, when the proposed scheme applied to JPEG images, the 8×8 BDCT coefficient 2-D array is replaced by the JPEG coefficient (JPEG quantized 8×8 BDCT coefficient) 2-D array. It is noticed that JPEG steganographic tools embed data to the JPEG coefficients.

2.2. Prediction-error

In [2], a prediction-error image is used to reduce the influence caused by the image content and enhance the noise introduced by data hiding. The effectiveness of the prediction-error images on steganalysis has been demonstrated in [2]. In this proposed steganalyzer, we use a different prediction scheme.

x	a
b	c

Figure 2. Prediction context

The prediction context is shown in Figure 2, i.e., we need to predict the value of x using the values of its neighbors a, b, and c. The prediction-error is given by

$$\hat{x} = x - sign(x) \cdot \{|a| + |b| - |c|\}. \quad (1)$$

Experimental investigation shows that compared to that used in [2] and [8], this prediction scheme can further improve steganalysis performances on texture images.

2.3. Discrete wavelet transform (DWT)

Wavelet decomposition has demonstrated its efficiency in steganalysis [1, 2]. For instance, the use of three-level wavelet decomposition has been justified in [2]. To balance steganalysis capability and computational complexity, we only conduct one-level wavelet decomposition in this scheme. If we consider the image, the BDCT coefficient 2-D array, or the JPEG coefficient 2-D array as LL_0, we have five subbands for one-level DWT decomposition. Compared to three-level DWT decomposition, the number of subbands reduces to 38%. Also, the Haar wavelet is used due to its simplicity.

2.4. Moments and marginal moments

The moments of the CF's are defined as:

$$M_n = \frac{\sum_{i=1}^{N/2} x_i^n |H(x_i)|}{\sum_{i=1}^{N/2} |H(x_i)|}, \quad (2)$$

where $H(x_i)$ is the CF component at DFT frequency x_i, N is the total number of different value level in a subband under consideration, i.e., DFT tap length, and $n = 1,2,3$.

For each 2-D array, we generate four second-order histograms (also referred to as 2-D histograms [8]) with the following separations $(\rho,\theta) = \{(1,0),(1,-\pi/2),(1,-\pi/4),(1,\pi/4)\}$, which are called horizontal, vertical, diagonal, and minor diagonal 2-D histograms, respectively. The two marginal moments of 2-D CF's are calculated by

$$M_{u,n} = \frac{\sum_{j=1}^{N/2}\sum_{i=1}^{N/2} u_i^n |H(u_i,v_j)|}{\sum_{j=1}^{N/2}\sum_{i=1}^{N/2} |H(u_i,v_j)|}, M_{v,n} = \frac{\sum_{j=1}^{N/2}\sum_{i=1}^{N/2} v_j^n |H(u_i,v_j)|}{\sum_{j=1}^{N/2}\sum_{i=1}^{N/2} |H(u_i,v_j)|}, \quad (3)$$

where $H(u_i,v_j)$ is the 2-D CF component at DFT frequency (u_i,v_j), N is the total number of different value level in a 2-D array under consideration, and $n = 1,2,3$.

2.5. Classification

Support vector machine (SVM) is used as classifier in our experimental work. To demonstrate the performance of a trained classifier, the area under the receiver operating characteristics (ROC) curve (AUC) [12] is used.

3. EXPERIMENTS AND RESULTS

798 texture images of 512×512 are used in our experimental investigation. Among these images, 578 were downloaded from the Internet [13, 14] and 220 were taken by a member of our research group. All these images are raw images.

Various experimental works have been conducted and the success of the proposed steganalyzer for texture images has been verified. Constrained by space, experiments reported in this paper focus on detecting generic LSB replacement, generic LSB matching [7], spread spectrum image steganography (SSIS) [15], generic quantization index modulation (QIM) [16], OutGuess [9], F5 [10], and MB1 [11]. When applying those JPEG steganographic tools, in order to avoid the effect on steganalysis caused by JPEG double compression, we deliberately ensure that the difference between a cover image and its associated stego image is caused only by data embedding. The parameters used in our experiments are shown in Table 1.

Table 1. Parameters of steganographic methods

Methods	Steganographic tools	Parameters
(1) LSB-R	Generic LSB Replacement (*)	0.1bpp
(2) LSB-M	Generic LSB Matching (*)	0.1bpp
(3) SSIS	SSIS	0.1bpp, SNR=40dB
(4) QIM	Generic QIM	0.1bpp, step=5
(5) OutGuess	OutGuess	0.1bpc (**)
(6) F5	F5	0.1bpc
(7) MB1	MB1	0.1bpc, step=2

* both the embedding position and the to-be-embedded data are randomly selected

** bits per non-zero AC (alternating current) coefficient

For comparison purpose, we generate features from the cover and the stego images with Farid's method [1] (hereinafter Farid's), Shi et al.'s method [2] (hereinafter Shi et al.'s), and our proposed method (hereinafter Proposed) to train and test the SVM classifier. For JPEG images, we also generate features with Chen et al.'s method [8] (hereinafter Chen et al.'s).

The SVM codes in MATLAB are downloaded from [17] and the polynomial kernel with degree 2 is used. At the training stage, 5/6 randomly selected cover/stego image pairs are used. The remaining 1/6 pairs are used for test. To eliminate the effect caused by image selection in training and test, we conduct 20 independent and random experiments and then average the experimental results.

3.1. Detecting data hidden in raw images

Firstly we conduct experiments on detecting data hidden in raw texture images (TXT in Table 2). To evaluate the performance of the proposed scheme on general smooth images, we also conduct experiments on the CorelDraw image dataset (CD in Table 2). The average AUC's of these experiments are reported in Table 2. Obviously, our proposed scheme outperforms Farid's and Shi et al.'s by a significant margin on texture images and also outperforms those two steganalyzers on CorelDraw images.

Table 2. Results on texture and CorelDraw images (AUC); TXT means texture images; CD means CorelDraw images

Steg. Methods	Farid's		Shi et al.'s		Proposed	
	TXT	CD	TXT	CD	TXT	CD
(1) LSB-R	0.5483	0.7168	0.5849	0.9790	0.8173	0.9914
(2) LSB-M	0.5467	0.7224	0.5727	0.9782	0.8017	0.9916
(3) SSIS	0.9802	0.9920	0.9089	0.9932	0.9853	0.9943
(4) QIM	0.9831	0.9943	0.8388	0.9935	0.9914	0.9945

3.2. Detecting data hidden in JPEG images

Our proposed scheme works not only on raw images, but also on JPEG images. Shown in Table 3 are the results on detecting JPEG steganographic methods 5, 6, and 7 applied to 798 JPEG texture

images. For comparison purpose, we also give the performance of method Chen et al.'s. It can be seen that the performance of our proposed scheme is not only much better than that of Farid's and Shi et al.'s, but also better than that of Chen et al.'s.

Table 3. Results on 798 JPEG texture images (AUC)

Steg. Methods	Farid's	Shi et al.'s	Chen et al.'s	Proposed
(5) OutGuess	0.6968	0.5709	0.8693	0.9382
(6) F5	0.5250	0.5517	0.7211	0.7481
(7) MB1	0.5302	0.5305	0.6631	0.7566

3.3. Demonstrating the effect of rake transform

The proposed scheme uses 270-D features. The first 54 feature components are derived from the spatial representation. The second, third, fourth, and fifth are derived from BDCT representation with block size 2×2, 4×4, 8×8 (JPEG coefficient 2-D array for JPEG images), and 16×16, respectively. Experimental results using part of and all of these features are given in Table 4. In this table, "54-D" means only the first 54 feature components (from the spatial representation) are used, "108-D" means the first and the second 54 feature components (from the 2×2 BDCT) are used, and so on.

It is observed that each of the BDCT's makes contribution to our proposed steganalyzer. Moreover, the more the BDCT (up to 16×16) included, the better the steganalysis performance. For raw images, it seems that the steganalyzer gives similar performance with 16×16 BDCT to that without 16×16 BDCT. However, 16×16 BDCT does enhance the steganalyzer's performance on JPEG images significantly.

Table 4. Results on 798 texture images (AUC)

Steg. Methods	54-D	108-D	162-D	216-D	270-D
(1) LSB-R	0.6665	0.7163	0.7635	0.8129	0.8173
(2) LSB-M	0.5877	0.6327	0.7196	0.7931	0.8017
(3) SSIS	0.8423	0.8874	0.9604	0.9829	0.9853
(4) QIM	0.7974	0.8579	0.9098	0.9900	0.9914
(5) OutGuess	0.5459	0.5542	0.5794	0.8976	0.9382
(6) F5	0.5345	0.5370	0.5474	0.6758	0.7481
(7) MB1	0.5368	0.5275	0.5504	0.6531	0.7566

4. SUMMARY

In this paper, we have proposed a novel universal steganalysis scheme and demonstrated the effectiveness of this scheme. This paper is summarized as follows:

(1) The difficulty in steganalyzing texture images has been reported and analyzed.

(2) The proposed universal steganalysis scheme greatly improves the capability in detecting data hidden in raw texture images. It also works quite well on general smooth images, e.g., the CorelDraw images. Our experiments have shown this scheme outperforms the state-of-the-art universal steganalyzers on both texture images and general smooth images. Therefore, it is an effective universal steganalyzer.

(3) The proposed steganalysis framework combines the features from the spatial representation and rake transform (referred to as block discrete cosine transforms with a set of various block sizes) representation. It also combines features from the 1-D characteristic function and features from the 2-D characteristic function.

(4) The proposed steganalysis scheme uses moments of 1-D characteristic functions and discrete wavelet transform, which are used in [2, 8], and marginal moments of 2-D characteristic functions, which are used in [8]. Furthermore, the proposed scheme utilizes features derived from rake transform, which make the proposed scheme more effective than that in [2] and [8].

(5) The prediction method used in this proposed steganalysis scheme, compared to that used in [2] and [8], also improves steganalysis performances.

(6) In addition, one-level DWT decomposition greatly reduces the computational complexity by reducing feature dimensionality compared to three-level DWT decomposition used in [2] and [8].

REFERENCES

[1] H. Farid, "Detecting hidden messages using higher-order statistical models", *International Conference on Image Processing*, Rochester, NY, USA, 2002.

[2] Y. Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen, "Steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network", *IEEE International Conference on Multimedia & Expo 2005*, Amsterdam, Netherlands, 2005.

[3] J. K. Hawkins, "Textural properties for pattern recognition", *Picture Processing and Psychopictoris*, B. C. Lipkin and A. Rosenfeld (editors), Academic Press, New York, 1970, pp 347-370.

[4] Y. Q. Shi, C. Chen, and W. Chen, "Steganalysis for texture images". http://www-ec.njit.edu/~shi/tech-rpt/s-c-c-2004.pdf.

[5] http: //www.corel.com/.

[6] R. Bohme, "Assessment of steganalytic methods using multiple regression models", *Information Hiding Workshop 2005*, Barcelona, Spain, June, 2005.

[7] A. D. Ker, "Steganalysis of LSB matching in grayscale images", IEEE Signal Processing Letters, 2005, 12(6): 441-444.

[8] C. Chen, Y. Q. Shi, and W. Chen, "Statistical moments based universal steganalysis using JPEG 2-D array and 2-D characteristic function", *IEEE International Conference on Image Processing 2006*, Atlanta, GA, USA, October, 2006.

[9] N. Provos, "Defending against statistical steganalysis," *10th USENIX Security Symposium*, Washington DC, USA, 2001.

[10] A. Westfeld, "F5 a steganographic algorithm: high capacity despite better steganalysis," *4th International Workshop on Information Hiding*, Pittsburgh, PA, USA, 2001.

[11] P. Sallee, "Model-based methods for steganography and steganalysis", *International Journal of Image and Graphics*, 5(1): 167-190, 2005.

[12] T. Fawcett, "Roc graphs: notes and practical considerations for researchers". http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf.

[13] http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

[14] http://www.cssip.uq.edu.au/staff/meastex/meastex.html.

[15] L. M. Marvel, C. G. Boncelet, Jr, and C. T. Retter, "Spread spectrum image steganography," *IEEE Trans. on Image Processing*, 8, pp. 1075--1083, August 1999.

[16] B. Chen and G. W. Wornell, "Digital watermarking and information embedding using dither modulation," *Proceedings of IEEE MMSP 1998*, pp273 – 278.

[17] C. C. Chang and C. J. Lin, LIBSVM: A library for support vector machines, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.