Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Alpha-trimmed Image Estimation for JPEG Steganography Detection

Mei-Ching Chen*, Sos S. Agaian†, C. L. Philip Chen†, and Benjamin M. Rodriguez‡
*†Department of Electrical and Computer Engineering
The University of Texas at San Antonio, San Antonio, TX, U.S.A.
‡Space Department, Johns Hopkins University Applied Physics Laboratory
Laurel, MD, U.S.A.
*axf710@my.utsa.edu

***Abstract*—In information security, steganalysis has been an important topic since evidences first indicated steganography has been used for covert communication. Among all digital files, numerous devices generate JPEG images due to the capability of compression and compatibility. A large number of JPEG steganography methods are also provided online for free usage. This has spawned significant research in the area of JPEG steganalysis. This paper introduces an image estimation technique utilizing the alpha-trimmed mean for distinguishing clean and steganography images. The hidden information is considered additive noise to the image. The alpha-trimmed method estimates steganographic messages within images in the spatial domain and provide flexibility for classifying various steganography methods in the JPEG compression domain. For three JPEG steganography methods along with three embedding message files applied to an image data set, the proposed method results in better separability between clean and steganographic classes. The results are based on comparisons between the presented method and two existing methods in which classification accuracies are increased by as much as 32%.**

***Index Terms*—Alpha-trimmed mean, image estimation, JPEG steganalysis, feature generation**

I. Introduction

Information security is and will continue to be a serious issue. Digital steganography has been one of the main vehicles used to secure data. Secret information is imperceptibly hidden within signals with the use of steganography. Signals containing enclosed messages are stored or transmitted through public channels without indication that pertinent information is hidden. On the other hand, behaviors of computer and cyber crime which consider steganography as a means of concealment lead to the problem of steganalysis [1].

The goal of image steganography detection is to determine whether a given image potentially contains secret data. For the problem of image steganalysis, approximation techniques can be used to resolve certain characteristics of an image in order to determine the existence of anomalies. This includes estimating anomalies in the image pixel values or coefficient values in the transform domain. The predicted pixel values/coefficients along with/without the original values can be used for generating features that are capable of separating input images into various categories. Related issues include detecting the existence of steganographic content, identification of the steganography method being used, extraction of the covert message, etc [2]. Among digital files, there are numerous sources that generate digital images. Furthermore, a majority of the devices create and store images as JPEG file types, a popularly used compressed image file format [3]. Due to a large number of online freeware generating steganography files with JPEG images, it is necessary to properly detect three forms of JPEG embedding methods:

- steganographic messages hidden within header files
- steganographic messages hidden within coefficients
- steganographic messages hidden within footers

This paper focuses on steganography detection of JPEG images, in which steganography methods embed the secret message within JPEG coefficients. Due to the characteristics of JPEG images, information hiding in JPEG coefficients is disseminated throughout the image in spatial domain pixel values without visually distorting the image. Hence, the hidden messages are considered additive noises within the spatial domain. This is the basis for developing an approximation technique for steganography images.

In the existing image feature generation methods for steganalysis, approximation techniques used for image pixel value or coefficient estimations are based on cropping in the spatial domains [4], [5], regression in the wavelet domains [6] and coefficient comparisons in the JPEG domain [7]. This paper presents a spatial domain estimation technique, the alpha-trimmed mean filter estimation. This method provides small amounts of noise estimation disseminated throughout the spatial domain and concentrated in the low and mid band coefficients in the JPEG quantized DCT blocks. Statistics are applied to both the original images and the predicted images for calculating a set of features. These statistics include a global histogram, individual histograms of low frequency coefficients, coefficient frequencies, coefficient variation, blockiness, and co-occurrence matrix of the coefficients [4].

The paper is organized as follows. Section II gives background knowledge of two feature generation methods, DCT features [4] and Markov features [7], as well as alpha-trimmed mean [8] which will be used to estimate a given image. The proposed method including image estimation and statistical measurements for generating features is described in Section III. Section IV illustrates the classifier utilized here [9]. In addition, this section also describes cross validation

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

for performance analysis. Section V shows and compares classification accuracies between the proposed method and two existing feature generation techniques with three JPEG steganography methods, F5 [10], Outguess v0.2 [11], and Steghide v0.5.1 [12], ensuing the conclusion and discussion of possible future work in Section VI.

II. Related Work

This section consists of brief descriptions regarding the alpha-trimmed mean which will be used to estimate a given image for steganalysis. Two existing feature generation methods are also described which are used for comparing the performance of the proposed feature generation method. The measure of performance is described in Section V.

A. Alpha-trimmed Mean

For an ascending sorted vector $\mathbf{x} = [x_1, x_2, \ldots, x_P]$ with length $P = 2m+1$ for $m \in \mathbb{N}$, the alpha-trimmed mean value of $\mathbf{x}$ is defined as in (1), where $t = \lfloor \alpha P \rfloor$ and $0 \leq \alpha \leq 0.5$ [8].

$$\mu_\alpha = \frac{1}{P-2t} \sum_{i=t+1}^{P-t} x_i \tag{1}$$

When $\alpha = 0$, μ_α implies the average of x_i; when $\alpha = 0.5$, μ_α indicates the median of x_i. Note that if $P = 2m$ for $m \in \mathbb{N}$ and when $\alpha = 0.5$, μ_α is the median of x_i, which is the average of the two values around the center [8].

B. DCT Features

For JPEG steganography detection, Fridrich generates 23 features using first order and second order statistics [4]. A given JPEG image $\mathbf{J}_1$ is decompressed into the spatial domain first. The image is then cropped by 4 pixels in each direction and recompressed with the same quantization table used in decompressing $\mathbf{J}_1$ to obtain $\mathbf{J}_2$. This is an approximation technique used to estimate altered pixel and coefficient values. A set of vector functions F is applied to $\mathbf{J}_1$, $\mathbf{J}_2$, and corresponding decompressed image pixel values. The final feature f corresponding to F is obtained from an L_1 norm calculating the difference between the function output of the original and the calibrated image. The L_1 norm is defined for a vector/ matrix as a sum of absolute values of all vector/matrix elements.

C. Markov Features

Shi et al [7] developed a set of features to detect JPEG steganography. The features generated view the differences in the JPEG 2-D array with Markov random process. According to the theory of random process, the transition probability matrix is used to characterize the Markov process. The features are derived from the transition probability matrix. In order to achieve an appropriate balance between steganalysis capability and computational complexity, a so-called one-step transition probability matrix is utilized. In order to further reduce computational cost by reducing the dimensionality of feature vectors, a thresholding technique is used. This results in a 324 dimensional feature vector, where the threshold used for range of coefficients is 4.

III. Image Estimation Feature Generation

For the problem of image steganalysis, in order to determine if anomalies in an image exist, an approximation technique can be used. One way is to approximate the characteristics of an image, such as image pixel values or coefficient values in transform domains. The original and/or predicted pixel values/coefficients along with/without some statistical measurements can be utilized for generating features that are able to separate different categories of input images. Fig. 1 shows a generic procedure of the technique.

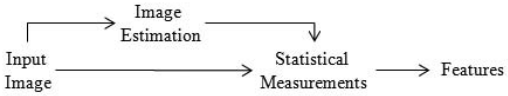

Fig. 1. A generic image estimation feature generation method

The idea of image estimation is to predict the pixel values which may be modified in an imperceptible fashion. The characteristics of images, called image features, can be measured by a set of statistics. The features generated are expected to distinguish input images as either clean or steganography classes.

A. Alpha-trimmed Mean Filter Estimation

JPEG steganography methods embed secret messages within JPEG coefficients resulting in hidden information disseminated throughout the image spatial domain pixel values without visually distorting the image. In this paper the secret message is considered additive noise to an image which leads to approximating the steganographic content in the spatial domain. The alpha-trimmed mean filter estimation is introduced here for steganography image estimation in the spatial domain.

The method divides the input image into blocks, either non-overlapped or overlapped, of size $u \times v$. The alpha-trimmed mean is applied to each block, generating a reduced intermediate image by averaging a portion of or all pixel values within the block. Suppose the image is approximated in a shrinking manner, interpolation techniques may be applied to this intermediate image. The output image is the same size of the input image, in which output values are approximations to the original values, see Fig. 2.

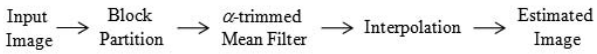

Fig. 2. Alpha-trimmed mean filter estimation process

When the image is re-expanded to its original size with an interpolation technique, a smoothing effect occurs. The blockiness of 8×8 blocks in a JPEG image is reduced. Note that a steganography image is imperceptibly different from its original clean state. The original input image and the predicted image derived from the image estimation process are used for deriving the features. In order to show the importance of image estimation, the statistical measurements initially introduced

in [4] are used along with the proposed estimation technique. This generates a set of 23 features, $\mathbf{f} = [f_1, f_2, \ldots, f_{23}]$, for classification between clean and steganography images.

B. Statistical Measurements

In the above technique, the estimated image in the spatial domain is to be saved as a JPEG image using the same quantization as the original input JPEG image. This results in a comparison of JPEG coefficients of the original input image and the newly created image. The idea of the comparison by subtracting the original coefficients with the estimated coefficients is to estimate coefficients that may have been manipulated which can be measured by a set of statistics [4] listed below. There are a total of 23 features, including 20 are derived from five functions, F_1 to F_5, using (2) calculated directly with coefficients. In (2), $\mathbf{J}_1$ indicates the JPEG coefficient values of original input image, while $\mathbf{J}_2$ would be the case of estimated input images, and f is the derived feature after calculating the L_1 norm between the functions of the original and the predicted.

$$f = \|F(\mathbf{J}_1) - F(\mathbf{J}_2)\|_{L_1} \tag{2}$$

- F_1: This function derives the first order statistic, i.e., the histogram, of all JPEG coefficients. This results in one feature value, f_1.
- F_2: This function obtains individual histograms of the first five AC coefficients, which are at location 2 to 6, after the zigzag procedure, as shown in Fig. 3, for each coefficient block. Only histograms of low frequency JPEG coefficients are used for deriving features. This is because histograms of coefficients at medium and higher frequencies are usually statistically unimportant due to the small number of non-zero coefficients. This results in 5 features, $f_2, \ldots, f_6$.

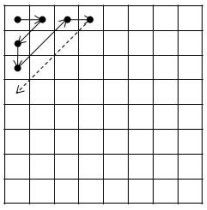

Fig. 3. The zigzag process of an 8 × 8 block

- F_3: This function counts the number of occurrences of all AC coefficients having values in the range from -5 to 5, inclusively. This results in a total of 11 features, $f_7, \ldots, f_{17}$.
- F_4: This function measures the variation between neighboring blocks within the image. The sum of differences between coefficients is calculated at the same location in each pair of neighboring blocks, both by rows and columns. The symbol $\mathbf{M}$ denotes the image in spatial domain, $|\mathbf{M}_r|$ is the number of blocks per row, and $|\mathbf{M}_c|$ is the number of blocks per column. This results in one feature value, f_{18}.

$$F_4(\mathbf{M}) = \frac{V_r + V_c}{|\mathbf{M}_r| + |\mathbf{M}_c|}$$

where

$$V_r = \sum_{i,j=1}^{8} \sum_{k=1}^{|\mathbf{M}_r|-1} \left| d_{ij}(\mathbf{M}_r(k)) - d_{ij}(\mathbf{M}_r(k+1)) \right|$$

and

$$V_c = \sum_{i,j=1}^{8} \sum_{k=1}^{|\mathbf{M}_c|-1} \left| d_{ij}(\mathbf{M}_c(k)) - d_{ij}(\mathbf{M}_c(k+1)) \right|$$

- F_5: This function measures the discontinuities along the 8 × 8 block boundaries, i.e., the blockiness of an image, in spatial domain. The symbol c indicates a pixel value in a decompressed image. Two features, f_{19} and f_{20}, are calculated from the sum of differences between border pixel values in each pair of neighboring blocks, both by rows and columns, with $\alpha = 1, 2$. The symbol M and N indicate the number of rows and the number of columns of decompressed images, respectively.

$$F_5(\mathbf{M}) = \frac{B_r + B_c}{N\lfloor (M-1)/8 \rfloor + M\lfloor (N-1)/8 \rfloor}$$

where

$$B_r = \sum_{i=1}^{\lfloor (M-1)/8 \rfloor} \sum_{j=1}^{N} \left| c_{8i,j} - c_{8i+1,j} \right|^{\alpha}$$

and

$$B_c = \sum_{j=1}^{\lfloor (N-1)/8 \rfloor} \sum_{i=1}^{M} \left| c_{i,8j} - c_{i,8j+1} \right|^{\alpha}$$

- f_{21}, f_{22}, f_{23}: The features are derived from co-occurrence matrices $C_{s,t}$ of neighboring JPEG coefficients [4], [13]. $\delta(\cdot,\cdot)$ compares the two parameters to see if they are equal. If yes, then the number counts, i.e., $\delta(\cdot,\cdot)$ = 1; otherwise, $\delta(\cdot,\cdot)$ = 0.

$$\begin{aligned} f_{21} &= D_{0,0} \\ f_{22} &= D_{0,1} + D_{1,0} + D_{-1,0} + D_{0,-1} \\ f_{23} &= D_{1,1} + D_{1,-1} + D_{-1,1} + D_{-1,-1} \end{aligned}$$

where $D_{s,t} = C_{s,t}(\mathbf{J}_1) - C_{s,t}(\mathbf{J}_2)$,

$$C_{s,t}(\mathbf{J}) = \frac{C_r + C_c}{|\mathbf{M}_r| + |\mathbf{M}_c|}$$

$$C_r = \sum_{i,j=1}^{8} \sum_{k=1}^{|\mathbf{M}_r|-1} \delta\left(s, d_{ij}(\mathbf{M}_r(k))\right) \delta\left(t, d_{ij}(\mathbf{M}_r(k+1))\right)$$

$$C_c = \sum_{i,j=1}^{8} \sum_{k=1}^{|\mathbf{M}_c|-1} \delta\left(s, d_{ij}(\mathbf{M}_c(k))\right) \delta\left(t, d_{ij}(\mathbf{M}_c(k+1))\right)$$

IV. Classification and Validation

A. Alpha-trimmed Standardization

The generated features from Section III lie in different dynamic ranges. This causes an unfair experimental condition, i.e., observations with initially large ranges have the potential to outweigh observations with initially smaller ranges. Hence, the problem can be overcome by preprocessing the features so that their values lie within similar ranges. The standardization method ensures that the data lies in the same dynamic range prior to classification. Each feature in this method is separately standardized by subtracting its mean and dividing by the standard deviation as follows:

$$\hat{\mathbf{f}} = \frac{1}{\sigma_\alpha}(\mathbf{f} - \mu_\alpha \cdot \mathbf{j})$$

where μ_α is alpha-trimmed mean, σ_α is its deviation for each feature from all of the available instances, and $\mathbf{j}$ is a vector of ones having the same length as the feature vector $\mathbf{f}$. Using alpha-trimmed mean with the standardization allows the removal of outliers without additional outlier processing [14].

B. Neural Network Classifier

A neural network classifier provided by Matlab is utilized here for nonlinear classification [9], [15], as shown in Fig. 4.

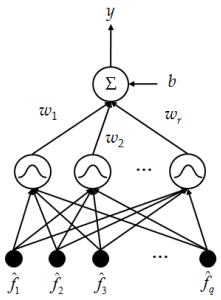

Fig. 4. A neural network model with Gaussian kernels

The input of the trained neural network classifier are the standardized feature vector $\hat{\mathbf{f}}$ of length q. The output y is the classification result indicating which class the input $\hat{\mathbf{f}}$ belongs to. For each node in the first layer network, Gaussian kernels ϕ_i with centers and spreads are applied. A weighted linear mapping in the second layer makes the classification decision with a threshold b. The model equation is defined in (3).

$$y = b + \sum_{i=1}^{r} \omega_i \phi_i(\hat{\mathbf{f}}) \tag{3}$$

C. Cross Validation

There are two main types of cross validation, random cross validation and k-fold cross validation. The former randomly divides the data set into subsets while the latter separates the data set into k mutually exclusive subsets (folds) [16]. As can be seen in Fig. 5, a k-fold cross validation is sometimes called rotation estimation. The folds are of approximately equal size. The inputs are trained on the selected training data and tested on the test data selection. The cross validation estimation of accuracy is the overall number of correct classifications divided by the number of instances in the data set. The accuracy estimate is the average accuracy for k mutually exclusive subsets.

Data Subsets	Entire Data Set			
1	Testing	Training		
2	Training	Testing	Training	
3	Training		Testing	Training
⋮	⋱	⋱		⋱
k	Training			Testing

Fig. 5. k-fold cross validation

Kohavi [16] has shown through experimental results on artificial data and theoretical results in restricted settings, that selecting a good classifier from a set of classification model, 10-fold cross validation may be better than the more expensive leave-one-out cross validation. In this research, this procedure will be adopted and carried out for all experimental results.

V. Computer Simulation

The clean image data set used in the experiment is downloaded from the website Break Our Watermarking System, 2nd edition [17], which contains 10,000 512 × 512 images. Two hundred images are randomly selected, converted from PGM uncompressed image file format into JPEG file format with a compression ratio of 75 for a fair comparison. For creating steganography images, three sizes of stego message text files, 0.21 KB, 0.32 KB, and 1.04 KB, as well as three popular JPEG steganography methods, F5 [10], Outguess v0.2 [11], and Steghide v0.5.1 [12], are used. Note that a file size of 1.04 KB contains the content of approximately one half page of a novel.

TABLE I
Classification Accuracies (%) for F5

	Observed					
	DCT Features [4]		Markov Features [7]		Proposed Method ($\alpha = 0$)	
	Stego	Clean	Stego	Clean	Stego	Clean
Stego 0.21KB	45.1	54.9	46.2	53.8	65.4	34.6
Clean	53.2	46.8	51.4	48.6	35.6	64.4
Stego 0.32KB	56.5	43.5	56.1	43.9	72.6	27.4
Clean	40.4	59.6	39.2	60.8	24.3	75.7
Stego 1.04KB	85.0	15.0	66.1	33.9	91.3	8.7
Clean	17.7	82.3	19.5	80.5	6.1	93.9

Using 10-fold cross validation, Table I, II and III show the confusion matrices as an average percentage of predicting clean images versus three steganography methods using the proposed features and two existing feature methods [4], [7]. In each of these experiments, there are 100 clean images and

TABLE II
CLASSIFICATION ACCURACIES (%) FOR OUTGUESS V0.2

	Observed					
	DCT Features [4]		Markov Features [7]		Proposed Features ($\alpha = 0$)	
	Stego	Clean	Stego	Clean	Stego	Clean
Stego 0.21KB	55.0	45.0	58.9	41.1	87.3	12.7
Clean	44.6	55.4	37.8	62.2	15.3	84.7
Stego 0.32KB	54.4	45.6	70.7	29.3	73.5	26.5
Clean	45.1	54.9	21.7	78.3	22.9	77.1
Stego 1.04KB	94.5	5.5	89.9	10.1	98.0	2.0
Clean	9.3	90.7	9.2	90.8	3.6	96.4

TABLE III
CLASSIFICATION ACCURACIES (%) FOR STEGHIDE V0.5.1

	Observed					
	DCT Features [4]		Markov Features [7]		Proposed Features ($\alpha = 0$)	
	Stego	Clean	Stego	Clean	Stego	Clean
Stego 0.21KB	68.7	31.3	72.4	27.6	87.3	12.7
Clean	28.1	71.9	14.1	85.9	16.8	83.2
Stego 0.32KB	67.3	32.7	75.9	24.1	76.9	23.1
Clean	27.6	72.4	35.4	64.6	19.7	80.3
Stego 1.04KB	72.7	27.3	87.1	12.9	93.6	6.4
Clean	22.7	77.3	9.8	90.2	4.8	95.2

100 steganography images used. The block size parameters for image estimations are $u = 2$ and $v = 2$.

From Table I, II and III, the use of proper image estimation methods allows an improvement in the capability of determining whether an image is clean or contains a secret message. In addition, a majority of steganalysis methods focus on a large amount of embedding data, the presented method is able to detect when a small amount of information is embedded.

VI. CONCLUSION

This paper presents the alpha-trimmed image estimation method for steganography detection on JPEG images. The idea considers the secret message being hidden in a cover image as additive noises. The method presented in Section III has the goal of accurately estimating JPEG coefficients that may contain hidden information resulting in higher classification accuracies. Three JPEG steganography methods, F5, Outguess v0.2, and Steghide v0.5.1, are used in the experimentation for testing two existing feature methods and the presented method. By using the proposed method along with one of the existing set of statistical measurements, the classification accuracies in the three steganography methods are all improved by as much as 32% higher classification accuracies. The alpha-trimmed mean filter estimation will be used in future work with additional statistical measure to improve the classification accuracies of steganography message files of various sizes.

ACKNOWLEDGMENT

The authors would like to thank the Department of Electrical and Computer Engineering at the University of Texas at San Antonio for supporting this research.

REFERENCES

[1] G. R. Gordon, C. D. Hosmer, C. Siedsma, and D. Rebovich, "Assessing technology, methods, and information for committing and combating cyber crime," 2003, http://www.ncjrs.gov/pdffiles1/nij/grants/198421.pdf.

[2] B. M. Rodriguez and G. L. Peterson, "Multi-class classification fusion using boosting for identifying steganography methods," in *Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, vol. 6974, 2008, pp. 697 407.1–697 407.10.

[3] M. Kharrazi, H. T. Sencar, and N. Memon, "Performance study of common image steganography and steganalysis techniques," *Journal of Electronic Imaging*, vol. 15, no. 4, pp. 041 104.1–041 104.16, 2006.

[4] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Proceedings of the Sixth International Workshop on Information Hiding, Lecture Notes in Computer Science*, vol. 3200. Springer Verlag, 2004, pp. 67–81.

[5] T. Pevný and J. Fridrich, "Merging markov and DCT features for multi-class JPEG steganalysis," in *Proceedings of SPIE/IS&T, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, 2007, pp. 650 503.1–650 503.13.

[6] H. Farid, "Detecting hidden messages using higher-order statistics," in *Proceedings of the International Conference on Image Processing*, vol. 2, 2002, pp. 905–908.

[7] Y. Q. Shi, C. Chen, and W. Chen, "A markov process based approach to effective attacking JPEG steganography," in *Proceedings of the Eighth International Workshop on Information Hiding, Lecture Notes in Computer Science*, vol. 4437, 2007, pp. 249–264.

[8] J. B. Bednar and T. L. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 1, pp. 145–153, 1984.

[9] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[10] A. Westfeld, "F5 - A steganographic algorithm," in *Proceedings of the Fourth International Workshop on Information Hiding, Lecture Notes in Computer Science*, vol. 2137, 2001, pp. 289–302.

[11] N. Provos, "Outguess 0.2," 2001, http://www.outguess.org/.

[12] S. Hetzl, "Steghide," 2003, http://steghide.sourceforge.net/.

[13] B. M. Rodriguez, S. S. Agaian, and J. F. Rodriguez, "Co-occurrence matrix feature vectors and cluster classification based steganalysis," in *INFORM*, 2004, p. 221.

[14] M. C. Chen, S. S. Agaian, C. L. P. Chen, and B. M. Rodriguez, "Steganography detection using RBFNN," in *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, vol. 7, July 2008, pp. 3720–3725.

[15] H. Demuth, M. Beale, and M. Hagan, *Raidal Basis Networks, Neural Network Toolbox™ 6 User's Guide*. The MathWorks Inc., 2009, http://www.mathworks.com/access/helpdesk/help/pdf_doc/nnet/nnet.pdf.

[16] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, vol. 2, 1995, pp. 1137–1145.

[17] P. Bas and T. Furon, "Break our watermarking system, second edition," 2008, http://bows2.gipsa-lab.inpg.fr/.