Steganalysis Using High-Dimensional Features Derived from Co-occurrence Matrix and Class-Wise Non-Principal Components Analysis (CNPCA)

Guorong Xuan[1], Yun Q. Shi[2], Cong Huang[1], Dongdong Fu[2], Xiuming Zhu[1], Peiqi Chai[1], and Jianjiong Gao[1]

[1] Dept. of Computer Science, Tongji University, Shanghai, P.R. China
grxuan@public1.sta.net.cn
[2] Dept. of Electrical & Computer Engineering, New Jersey Institute of Technology
Newark, New Jersey, USA
shi@njit.edu

Abstract. This paper presents a novel steganalysis scheme with high-dimensional feature vectors derived from co-occurrence matrix in either spatial domain or JPEG coefficient domain, which is sensitive to data embedding process. The class-wise non-principal components analysis (CNPCA) is proposed to solve the problem of the classification in the high-dimensional feature vector space. The experimental results have demonstrated that the proposed scheme outperforms the existing steganalysis techniques in attacking the commonly used steganographic schemes applied to spatial domain (Spread-Spectrum, LSB, QIM) or JPEG domain (OutGuess, F5, Model-Based).

Keywords: steganalysis, co-occurrence matrix, class-wise non-principal components analysis (CNPCA).

1 Introduction

This paper[1] addresses universal image steganalysis under the framework of pattern recognition. The steganalysis is the counterpart of steganography. The purpose of the steganalysis is to detect the hidden message, equivalently, to discriminate the stego-object from the non-stego-object. The steganalysis techniques proposed in the literature can be classified into two categories: the universal steganalysis which is designed to detect the hidden message embedded with various data embedding algorithms, and the specific steganalyis which is designed to attack a specific steganography technique.

Farid [1] proposed a universal steganalysis algorithm based on high-order statistical moments derived from high-frequency wavelet subbands. These statistics are based on decomposition of images with separable quadrature mirror filters. The high-frequency subbands' statistical moments are obtained as features for

[1] This research is supported partly by National Natural Science Foundation of China (NSFC) on the project (90304017).

Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283, pp. 49 – 60, 2006.

steganalysis. It can differentiate stego-images from non-stego (also referred to as cover) images with a certain success rate. In [2], Xuan et al. proposed a universal steganalysis approach, which selects statistical moments of characteristic functions of the test image, and all of their wavelet subbands as features. This steganalyzer outperforms [1] in general. In [3], Fridrich developed a steganalysis scheme specifically designed for attacking JPEG steganography. A set of well-selected features for steganalysis are generated from the statistics of the JPEG image and its calibrated version. This scheme outperforms [1] and [2] in attacking the JPEG steganography, such as OutGuess [4], F5 [5], and Model-based (MB) [6]. The feature extraction algorithm of the steganalyzer [3], however, is complicated and takes time.

The above mentioned steganalysis schemes [1, 2] are both based on the statistics of the histogram of wavelet subbands. (Note that the scheme [2] is partially based on histogram of given test image as well.) Histogram itself is known as the first order statistics. In [3], in addition to the first order statistics, histogram, the second order statistics such as co-occurrence in the JPEG coefficient domain are also used to extract features. However, only the co-occurrence counted from some modes of JPEG coefficients between neighboring blocks has been used. It is noted that the Markov chain was firstly used for steganalysis by Sullivan et al. [7]. There, they scan the whole image horizontally row-by-row and then calculate the empirical transition matrix, which is essentially something similar to the co-occurrence matrix. Since the dimensionality is extremely high (e.g., $256 \times 256 = 65,536$ for an 8-bit gray-level image), not all of elements of the matrix can possibly be used as features. Only some elements are selected. The authors of [7] select several largest probabilities along the main diagonal together with their neighbors, and then randomly select some other probabilities along the main diagonal as features, resulting in a 129-dimensional feature vector. Finally, supporting vector machine (SVM) is adopted in their scheme for classification. This technique, though successful to some extent for the detection of spread spectrum (SS) data hiding, does not perform well for attacking other steganography methods, in particular, for those JPEG steganographic methods. One of reasons is it has abandoned some useful information due to the random fashion of some features' selection.

Inspired by [7], this paper presents a new steganalysis scheme based on the high-dimensional features generated from the co-occurrence matrix. In this scheme, we propose to adopt high-dimensional features, hence using the adequate information of co-occurrence matrix, to capture the changes before and after the data embedding. The class-wise non-principal component analysis (CNPCA) is proposed to solve the classification problem in high-dimensional space. In addition to working on the gray-level co-occurrence matrixes for attacking steganographic methods in spatial domain, we also work on the co-occurrence matrixes associated with JPEG coefficient domain to attacking modern JPEG steganographic techniques, such as OutGuess [4], F5 [5], and MB [6]. Considering the 2-D nature of images, in either case, we consider the vertical, main-diagonal and minor-diagonal directions in addition to the horizontal direction when generating co-occurrence matrixes. Our extensive experiments have demonstrated that the proposed scheme performs better than the existing steganalysis schemes.

The rest of the paper is organized as follows. Section 2 describes the feature extraction from gray-level co-occurrence matrix. The CNPCA classification method

is introduced in Section 3. The scheme to attack JPEG steganography is described in Section 4. The experimental results are given in Section 5 and the paper is concluded in Section 6.

2 Feature Extraction from Gray-Level Co-occurrence Matrix

Gray-level co-occurrence matrix describes the co-occurrence of the various gray-levels at some specified spatial positions in an image. The gray-level co-occurrence matrix of the natural image tends to be diagonally distributed because the gray-levels of the neighbor pixels in natural images are often highly correlated. After the data embedding, however, the high-concentration along the main diagonal of gray-level co-occurrence matrix spreads because the high-correlations between the pixels in the original image have been reduced. This has been shown in [7].

The parameters of gray-level co-occurrence matrix in our scheme are chosen as follows. The gray-levels are 0-255 for 8-bits gray-level images. The gray-level co-occurrence matrix offset parameter d [8] is set to 1, namely, only the nearest neighborhoods are considered in our method. Four different directions are selected for gray-level co-occurrence matrix calculation [8], i.e., θ = 0°, 45°, 90° and 135°, respectively. We thus obtain four gray-level co-occurrence matrixes: G1, G2, G3, G4 from these four different directions, respectively. From these four matrixes, we generated the following resultant co-occurrence matrix, i.e.,

$$G = normal\left(G_1 + G_2 + G_3 + G_4\right) \tag{1}$$

where the operator *normal* represents average and normalization.

	-1	**0**	**1**
-1		1	
0	1		3
1		3	

Fig. 1. The co-occurrence matrix of the 1-D sequence [-1, 0, 1, 0, 1]

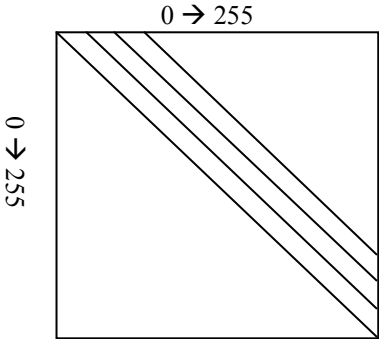

Fig. 2. Gray-level co-occurrence matrix (256×256)

According to [8], the co-occurrence matrix we generated is a symmetric matrix. That is, we not only count the occurrence of *a* with *b*, but also the occurrence of *b* with *a*. As a simple example, consider a 1-D signal [-1, 0, 1, 0, 1]. Its co-occurrence matrix is symmetric as shown in Fig. 1.

Considering the symmetry of the co-occurrence matrix, we adopt the elements of the main diagonal and a part of the upper triangle of the matrix, as shown in Fig. 2, to construct the feature vector in our proposed scheme. In our experimental work, as shown in Fig. 2, we use 1018-dmensional (256×4-6=1018□feature vectors. Statistically, the energy of the selected elements is about 50-70% of the whole upper triangle of the gray-level co-occurrence matrix. The selected feature vector, therefore, keeps most information of the gray-level co-occurrence matrix and is expected to be able to capture the changes caused by the data embedding process.

Let G_{ori} denote the gray-level co-occurrence matrix of the original cover image and G_{steg} the gray-level co-occurrence matrix of the stego image. Thus, $(G_{ori}-G_{steg})^2$ describes the energy differences between them, which is shown in Fig. 3. It is observed from Fig. 3 that the energy difference concentrates around the main diagonal of gray-level co-occurrence matrix. This observation together with the symmetry of co-occurrence matrix justifies our feature selection of the elements of gray-co-occurrence matrix as shown in Fig. 2.

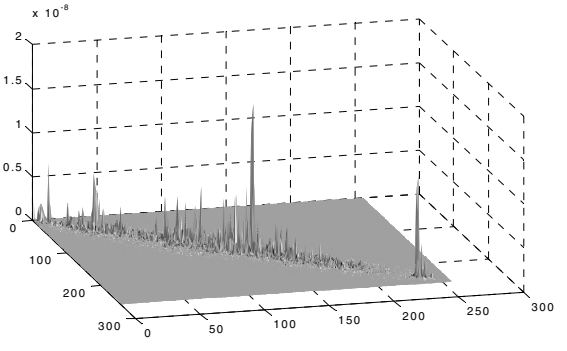

Fig. 3. The distribution of energy difference (refer to text)

If we adopt the Euclidean distance based Bayes classifier to classify the 1018-dimensional feature vectors, it would have been very hard to calculate the inverse covariance matrix because of the high dimensionality. To solve this problem, we propose the class-wise non-principal components analysis method.

3 Class-Wise Non-Principal Component Analysis (CNPCA)

The class-wise non-principal component analysis (referred to as CNPCA for short) [9] is to classify the samples based on the distances between the samples and the mean vectors of each class in the space spanned by the eigenvectors associated with the smallest eigenvalues of each class.

3.1 Definitions

Let $\boldsymbol{x}$ denote the n-dimensional random vectors in the k^{th} class, and assume that there are in total K different classes. When the eigenvalues of the covariance matrix generated from all of $\boldsymbol{x}$ are ranked from the largest to the smallest in an non-increasing order, the corresponding eigenvector matrix can be expressed as:

$$\Phi_k = \left(\Phi_k\right)_{n\times n} = \left[\Phi_{rk}, \Psi_{rk}\right]_{n\times n} \tag{2}$$

where the n is the dimensionality; the r, $(r \le n)$, is the number of eigenvectors associated with the largest eigenvalues; the $(n\text{-}r)$ is the number of eigenvectors associated with the smallest eigenvalues; the $\boldsymbol{\Phi_k} = \left(\Phi_k\right)_{n\times n}$ is the eigenvector matrix with all eigenvectors of the k^{th} class; the $\boldsymbol{\Phi_{rk}} = \left(\Phi_{rk}\right)_{n\times r}$ is the principal components matrix with all the r eigenvectors of the k^{th} class; the $\Psi_{rk} = \left(\Psi_{rk}\right)_{n\times(n-r)}$ is the non-principal components matrix with all, $(n\text{-}r)$, remaining eigenvectors of the k^{th} class; the k^{th} class' non-principal components $\boldsymbol{\Psi_{rk}}$ and principal components $\boldsymbol{\Phi_{rk}}$ are complementary to each other.

In CNPCA classification, given a test sample vector $\boldsymbol{y}$, its Euclidean distance to the mean vector of the k^{th} class in the subspace spanned by the $(n\text{-}r)$ class non-principal components is adopted as the classification criterion, referred to as CNPCA distance. The CNPCA distance of the vector $\boldsymbol{y}$ to the k^{th} class is defined as:

$$D_{rk} = \left\| \Psi_{rk}^{t}\left(y - M_k\right)\right\| \tag{3}$$

where D_{rk} stands for the Euclidean distances between the sample $\boldsymbol{y}$ and the mean of the k^{th} class, M_k, in the $(n\text{-}r)$ dimensional CNPCA space, $\boldsymbol{D_{rk}}$ can be represented by the class-wise non-principal components matrix $\boldsymbol{\Psi_{rk}}$. Obviously, there are two special cases. When $r=0$, CNPCA distance becomes the conventional Euclidean distance while when $r=n$, CNPCA distance equals to 0. Hence the case of $r>0$ and $r<n$ is usually used in CNPCA.

In summary, during the CNPCA classification, a given test sample $\boldsymbol{y}$ is firstly mapped into the $(n\text{-}r)$ non-principal components subspace of each class. The distances in these subspaces between $\boldsymbol{y}$ and the mean of each class are then calculated. Finally the $\boldsymbol{y}$ is classified to the k^{th} to which the CNPCA distance is the minimum, i.e.,

$$\hat{k} = \arg\min_{k}\left\{\ D_{rk}\ \right\}.$$

The number of r is an important parameter for CNPCA. It can be estimated by minimizing the classification error rate ε:

$$\hat{r} = \arg\min_{r}\left\{\ \varepsilon\left(D_{rk}\right)\ \right\}$$

3.2 Classification Procedure

Step 1: We first apply the K-L transform to the training samples of each class. The $(n\text{-}r)$ eigenvectors associated with the $(n\text{-}r)$ smallest eigenvalues are selected as the

dimension reduction matrix, Ψ_{rk} , for each class. The mean vector of each class, $\boldsymbol{M_k}$, is also calculated in this step.

Step 2: For testing sample $\boldsymbol{y}$, the CNPCA distances between the sample and each class, $\boldsymbol{D_{rk}}$, are calculated according to the following formula,.

$$D_{rk} = \left\| \Psi_{rk}^{t} \left(y - M_k \right) \right\| = \left(y - M_k \right) \Psi_{rk} \Psi_{rk}^{t} \left(y - M_k \right) \tag{4}$$

Step 3: The testing sample $\boldsymbol{y}$ is classified to the class to which the $\boldsymbol{y}$ has the minimum CNPCA distance $\boldsymbol{D_{rk}}$. In other words, the classification decision is made by:

$$\hat{k} = \arg\min_{k} \left\{ D_{rk} \right\} \tag{5}$$

3.3 CNPCA vs. PCA

The concept of CNPCA classification is quite different from that of the conventional PCA classification. While the CNPCA method utilizes the within-class information in each class effectively, the PCA (Principal Component Analysis) is a dimension reduction method for the whole set which averages the within-class distribution of each class. When the samples scatter within each class and cluster between classes, the PCA does not perform well. On the contrary, CNPCA describes the minimum variance directions for each class. It is very suitable to solve the problem of the scattering within each class and clustering between classes. Actually, image steganalysis is a typical two-class ("stego-image" and "non-stego-image") classification problem in which the samples scatter within class and cluster between classes. The content of the image database is very diverse. The samples, therefore, scatter within each class. On the other hand, because the embedding process must be invisible, the data embedding strength has to be small enough, which makes the samples cluster between classes. The CNPCA removes the principal components while keeps the non-principal components. The main purpose of doing so is to select features which are sensitive to "embedding" instead of "the image content itself" from the high dimensional space of gray-level co-occurrence matrix. For more detail, readers are referred to [9].

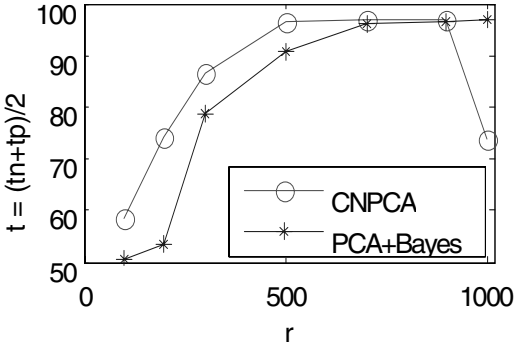

Fig. 4. Comparison of CNPCA and PCA+Bayes classifier (for image database refer to Section 5.1), embedding method is LSB with 0.1bpp), tn: true negative rate, tp: true positive rate

To verify this idea, we compare the performance of the proposed CNPCA classifier, and the PCA dimension reduction followed by a Bayes classifier in Figure 4.

4 Attacking JPEG Steganography

Since the JPEG (Joint Photographic Experts Group) format is the most dominant image format for image storage and exchange these days, the JPEG steganographic techniques have attracted more and more attentions. In JPEG steganographic techniques, secret message are embedded by modifying the quantized block-DCT coefficients of the cover image. Steganalyzing the JPEG steganography directly in JPEG block-DCT coefficient domain is more effective than doing it in image pixel domain. Therefore, we calculate the co-occurrence matrix in the block-DCT domain when attacking JPEG steganography.

The procedure of the feature extraction is as follows:

(1) Read in the quantized block-DCT coefficients from a given JPEG file.
(2) Expand the block-DCT coefficients of each 8×8 block into a 1-D vector $V_i(0,1,2,\ldots,63)$ in the zig-zag order [10], where i is the block index.
(3) Only keep the low frequency part of the 1-D vector, i.e., $V_i(1,2,\ldots,20)$. We do it in this way because most of the high frequency coefficients are quantized to zero (thus not much information will be lost), and few modern steganographic methods touch DCT DC coefficients. Since the magnitude of the block-DCT coefficients has a quite large dynamic range, we further clip the values of V_i to the range of $[-T, T]$, where T is a predefined threshold. Properly setting a threshold will not lose much information because the block DCT coefficients follow the generalized Laplacian distribution which has a very large peak around zero, and most of DCT AC coefficients are small. As an example, we have calculated the percentage of the block DCT AC coefficients which are below a given threshold T for the popularly used Lena image with Q-factor 80 in JPEG compression, and found that for $T=7$, 96.59% of block DCT AC coefficients fall into the interval of $[-T, T]$. Therefore, using an appropriate threshold only lose trivial information while reduce the computational complexity dramatically.
(4) Calculate the co-occurrence matrix G_i for each 1-D vector V_i.
(5) Calculate the global average co-occurrence matrix by using $G=\frac{1}{N}(\sum_{i=1}^{n}G_i)$, where N is the total number of the blocks in a JPEG image. We use the whole upper triangle of G as feature for steganalysis. Finally, CNPCA is employed as classifier for classification.

5 Experiments in Steganalysis

5.1 Attacking Steganography in Spatial Domain

We generate a hybrid image database composed of totally 3908 images, in which 1096 images are from CorelDraw [11] and the other 2812 images are from the UCSB web site [12]. In the experiments, we randomly select half of the images (1954

images） as the training samples and the other half as the testing samples. Three embedding methods: Cox et al.'s spread spectrum (SS), QIM, and LSB are used in the experiments and the embedding rate are set to be 0.3 bpp (bits per pixel), 0.1 bpp and 0.02 bpp, respectively. The experimental results are shown in Table 1 and Fig. 5.

Table 1 illustrates the performance comparison between the proposed scheme (from *r=500* to *r=900*) and the steganalysis methods proposed by Farid [1] and Sullivan et al. [2]. The detection rates shown in Table 1 are actually the average results of 10 times tests（the training and testing samples are randomly selected each time). As can be seen in Table 1, our proposed method outperforms Farid's and Sullivan's method for all the embedding methods (except QIM) at all the embedding rates. This superiority is obvious especially for the low embedding rate LSB0. (1 and 0.02 bpp).

Table 1. Detection accuracy comparison (tn: true negative; tp: true positive; t=(tn+tp)/2)

	Farid [1]			Sullivan et al. [7]			Proposed		
	tn	tp	t	tn	tp	t	tn	tp	t
SS	23	89	56	86	64	75	76	82	79
Qim	66	92	79	91	90	91	78	97	87
LSB(0.3bpp)	37	91	64	56	74	65	99	99	99
LSB(0.1bpp)	23	89	56	45	62	53	96	98	97
LSB(0.02bpp)	8	92	50	39	57	48	73	79	76

5.2 Selection of Dimensionality *r*

According to [13], in our proposed method, the detection rate *t* is defined as the arithmetic average of true positive rate and true negative rate. It is also referred to as

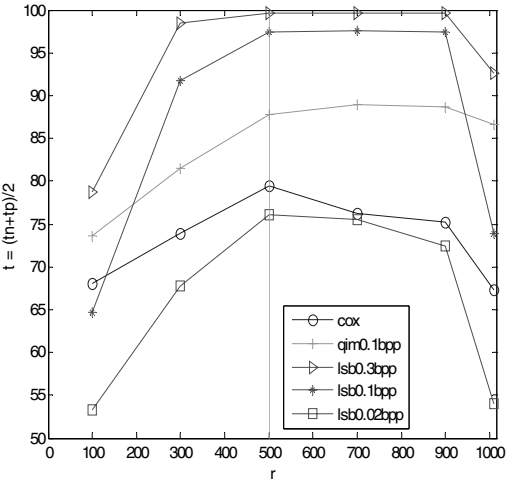

Fig. 5. Average detection rate as a function of the non-principal components dimensionality *n-r*

accuracy. It is a function of the non-principal components dimensionality (n-r). Generally speaking, it can keep high detection rate in a relatively wide range around the peak value. As shown in Fig. 5, the detection rates achieve their peak values when r is around 500 (at this point, the dimensionality of the non-principal components is 1018-500=518). The peak values keep almost constant till r=*900* (at this point, the dimensionality of the non-principal components is 1018-900=118).

5.3 Stability Study

To verify the stability of the detection rate, we repeat the test 10 times in this experiment. Each time, we randomly select half of the 3908 images as the training set and the other half as the testing set. The results (r is set to 500) are recorded and shown in Figure 6.

As we can see in Figure 6, the detection rates are quit stable for different training and testing sets.

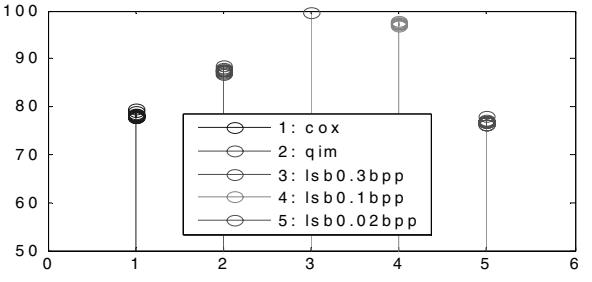

Fig. 6. Performance of the stability experiments (The numbers in horizontal axis stand for different embedding scheme and the vertical axis denotes the detection rate. Tests on each embedding method are repeated 10 times).

5.4 Attacking JPEG Steganography

In JPEG steganalysis, to avoid the influence of JPEG double compression on steganalysis performance, we use the 1096 uncompressed images from the CorelDraw database [11]. We firstly JPEG [14] compress all the images using Quality-factor 75 as the cover images. All the images are also embedded by separately using OutGuess [15], F5 [16], MB1 [17] (MB without deblocking operation) and MB2 (MB with deblocking operation) with the amount of embedded message as 1kB (i.e., 1024 bytes), 2kB and 4kB, respectively. The images have been cut to keep the central portion having a size 768x512 or 512x768. The central cut was conducted in the JPEG coefficient domain in order not to involve additional JPEG compression.

In the classification process, we randomly selected 896 (about 5/6 of) original images and the corresponding 896 stego-images for training and the remaining 200 pairs (about 1/6) of stego images and non-stego images for testing. For comparison purpose, we have also implemented the steganalysis schemes proposed by Farid [1], Xuan et al. [2] and Fridrich [3]. Then we apply them to the same set of images and with the same steganographic methods. The same training and testing procedures are used. All the

results are listed in Table 2. The experimental results reported here are the averages of the 10 times of random tests. The r parameter in CNPCA classification is selected as follows. For F5, the detection rate peak value appears at $r = \{3, 4, 5, 6, 7\}$. We select $r = 5$ in Table 2. For OutGuess, MB1 and MB2, the detection rate peak value appears at $r = \{10, 11, 12, 13, 14, 15]$. We select $r = 12$ in Table 2.

As can be seen in Table 2, the proposed scheme outperforms all the other steganalysis schemes in detecting these four JPEG steganographic techniques at all of these three different data embedding rate. The exceptions are in detecting F5 at embedding rates of 1kB and 2kB, i.e., the detection rates achieved by our proposed method is 1% and 2%, respectively, less than that achieved by [3]. In general, the detection rates achieved by the proposed scheme are comparable to or higher than that by Fridrich's method [3] while outperforms that by Farid's [1] and Xuan et al.'s [2] by a significant margin. As a whole, therefore, the proposed scheme outperforms the existing steganalysis methods.

Table 2. Performance comparison in JPEG steganalysis

		Farid			Xuan et al.		
		tn	tp	t	tn	tp	t
F5	1kB	51	54	53	62	58	60
	2kB	56	56	56	64	65	65
	4kB	68	53	60	85	77	81
Outguess	1kB	58	38	48	61	41	51
	2kB	61	39	50	67	61	64
	4kB	59	48	54	73	79	76
MB1	1kB	48	55	52	59	60	59
	2kB	52	53	53	71	69	70
	4kB	53	58	55	83	81	82
MB2	1kB	55	47	51	65	55	60
	2kB	49	58	53	75	64	69
	4kB	59	52	55	85	83	84
		Fridrich			Our proposed		
		tn	tp	t	tn	tp	t
F5	1kB	76	72	74	78	68	73
	2kB	86	87	87	84	85	85
	4kB	94	98	96	97	98	98
Outguess	1kB	91	87	89	98	98	98
	2kB	97	96	97	100	100	100
	4kB	98	97	98	100	100	100
MB1	1kB	66	65	66	90	84	87
	2kB	88	83	86	98	88	93
	4kB	91	88	90	99	99	99
MB2	1kB	64	61	62	89	82	86
	2kB	76	77	76	98	92	95
	4kB	88	80	84	100	99	99

5.5 Computational Complexity

Computational complexity is important to estimate a system's potential for real-time application. In this section, time cost of different steganalysis schemes is used as a main criterion for measuring computational complexity. In this experiment, we only test the time spent for feature extraction, which consumes most of the time, and classifier's parameters can be trained before actually use. Randomly select 100 JPEG images form the database described in 5.4 for testing the time cost, the result is show in Table 3. The experimental environment is: Intel Celeron(R) CPU 1.70GHz, memory 256MB, and Matlab 7.1 version.

Table 3. Time cost of features extraction

Steganalysis	Featrues dimention	Seconds of 100 images	Seconds per image
Farid[1]	72	657.85	6.58
Xuan[2]	39	406.46	4.06
Fridrich[3]	23	1159.20	11.59
Sullivan[7]	130	130.51	1.30
Proposed	120	130.30	1.30

As we can see from Table 3 that though more dimensions are used in our proposed scheme, the time cost is the least, while the scheme proposed in [3] has highest computational complexity, even though it uses only 23 features. From these experiments, one can observe that the proposed steganalysis method is effective both in high detection rate and low time cost, which lends itself for potential of real-time usage in practice.

6 Conclusions

1) We have proposed to use the high dimensional features generated from co-occurrence matrix in image pixel domain and in JPEG coefficient domain to capture the changes occurring to images before and after the data embedding.
2) Class-wise non-principal component analysis (CNPCA) is proposed to be utilized as classifier for steganalysis. The CNPCA classification overcomes the problems where the inverse of covariance matrix does not exist in pattern classification. It demonstrates good performance in tackling the problem caused by high dimensionality of feature vectors and/or the problem where between-class feature vectors cluster while within-class vectors scatter.
3) The experimental works have demonstrated improved performance in steganalysis, compared with the existing steganalyzers.

References

1. Farid H.: Detecting hidden messages using higher-order statistical models. Proceeding of the IEEE International Conference on Image Processing, Vol. II, New York, (2002) 905 - 908
2. Xuan, G., Shi, Y.Q., Gao, J., Zou, D., Yang, C., Zhang, Z., Chai, P., Chen, C., Chen, W.: Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. Information Hiding Workshop, Barcelona, Spain, June (2005) 262 – 277

3. Fridrich, J.: Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. 6th Information Hiding Workshop, Toronto, Canada (2004)
4. Provos, N.: Defending against statistical steganalysis. 10th USENIX Security Symposium, Washington DC, USA (2001)
5. Westfeld, A.: F5 a steganographic algorithm: High capacity despite better steganalysis. 4th International Workshop on Information Hiding, Pittsburgh, PA, USA (2001)
6. Sallee, P.: Model-based methods for steganography and steganalysis. International Journal of Image and Graphics, 5(1) (2005) 167-190
7. Sullivan, K., Madhow, U., Chandrasekaran S., Manjunath, B.S.: Steganalysis of spread spectrum data hiding exploiting cover memory. SPIE 2005, vol. 5681, (2005) 38 - 46
8. Haralick, R.M.: Textural features for image classification. IEEE Trans. Systems Man Cybernetics. SMC-3 (1973)
9. Xuan, G., Chai, P., Zhu, X., Yao, Q., Huang, C., Shi, Y.Q., Fu, D.: A novel pattern classification scheme: Classwise non-principal component analysis (CNPCA). International Conference on Pattern Recognition (ICPR), Hong Kong, August (2006)
10. Shi, Y.Q., Sun, H.: Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards. CRC Press, Boca Raton, FL. (1999)
11. http://www.corel.com
12. http://vision.ece.ucsb.edu/~Sullivan/Research_imgs/
13. Fawcett, T.:"ROC Graphs: Notes and Practical Considerations for Researchers", Tech Report HPL-2003-4, HP Laboratories. (2003)
(http:// home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf)
14. http://www.ijg.org/
15. http://www.outguess.org/
16. http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html
17. http://redwood.ucdavis.edu/phil/papers/iwdw03.htm