Multimed Tools Appl (2014) 72:313–330
DOI 10.1007/s11042-013-1373-4

Digital video steganalysis by subtractive prediction error adjacency matrix

Keren Wang · Jiesi Han · Hongxia Wang

Published online: 31 January 2013

Abstract Video has become an important cover for steganography for its large volume. There are two main categories among existing methods for detecting steganography which embeds in the spatial domain of videos. One category focuses on the spatial redundancy and the other one mainly focuses on the temporal redundancy. This paper presents a novel method which considers both the spatial and the temporal redundancy for video steganalysis. Firstly, model of spread spectrum steganography is provided. PEF (Prediction Error Frame) is then chosen to suppress the temporal redundancy of the video content. Differential filtering between adjacent samples in PEFs is employed to further suppress the spatial redundancy. Finally, Dependencies between adjacent samples in a PEF are modeled by a first-order Markov chain, and subsets of the empirical matrices are then employed as features for a steganalyzer with classifier of SVM (Support Vector Machine). Experimental results demonstrate that for uncompressed videos, the novel features perform better than previous video steganalytic works, and similar to the well-known SPAM (Subtractive Pixel Adjacency Model) features which are originally designed for image steganalysis. For videos compressed with distortion, the novel features perform better than other features tested.

Keywords Video steganalysis · Communication security · Steganography · SPAM · SPEAM

1 Introduction

Steganography is the art of hiding messages into cover objects such as images, texts, videos and network protocol packets. In order to make the stego object unperceivable, the sender

K. Wang
National Digital Switching System Engineering & Technological Research Center, Zhengzhou, China

K. Wang (✉) · J. Han
Science and Technology on Blind Signal Processing Laboratory, Chendu, China
e-mail: cfan662003@gmail.com

H. Wang
Southwest Jiaotong University, Chengdu, China

applies a mutually independent embedding operation to selected elements of the cover. Steganalysis is the art of detecting the existence or even determining the location, volume and extracting the content of hidden messages in various cover objects. When the existence of hidden messages is detected, the security of the steganographic system is believed to be destroyed. In this paper, the main concern lies in the detection of hidden messages embedded in videos.

Most embedding methods for videos are developed from those for images. A popular method under this paradigm is LSB (Least Significant Bit) matching[24, 27], which randomly increases or decreases pixel values by one to match the LSBs with the candidate message bits. Besides, Some other embedding methods, such as QIM (Quantization Index Modulation) [13, 19], and SS (Spread Spectrum) steganography[6, 8, 16] etc., are also introduced into spatial video steganography. Another kind of embedding method can be represented as MSU StegoVideo [17] which is a spatial video steganographic software got from the Internet. As videos are often coded before transmission to the receiver, and videos got from the Internet are always compressed in volume to save bandwidth, the robustness of steganographic methods in the spatial domain is very important for the restore of hidden messages. Most of those steganographic methods maintain robustness by embedding several copies of the original hidden messages into the cover videos.

Various methods have been designed to detect steganography in the spatial domain of videos. Kundur and Budhia [2] proposed a detection method based on collusion. It was also called TFA (Temporal Frame Averaging), which was commonly used in the research on watermark attacking. The estimated cover was got coarsely through collusion, and residuals between estimated frame and the stego frame were calculated. Kurtosis, entropy, and 25 % percentile of the residuals were utilized as features for steganalysis. In [3], Kurdur and Budhia further explained the basic theory and the effective condition of collusion. In [9], MoViSteg (Motion-based Video Steganalysis) was proposed by Jainsky et al. Motion interpolation was used to get a coarse estimation of cover object. Residuals between estimated copy and the stego one were then analyzed by ARE (Asymptotic Relative Efficiency), and adaptive threshold was adopted for the detection results. In [18], the local variance of the prediction error frame was calculated in size of 3×3. Gamma distribution was adopted to fit the distribution of the local variance, and two parameters of the distribution were extracted as the steganalytic features.

Besides temporal redundancy utilized in the steganalytic methods stated above, there is also spatial redundancy in the content of videos. Spatial averaging has been used for video steganalysis. In [23], 3×3 spatial average filtering was adopted to estimate the cover. Features used in [3] were then extracted for the steganalyzer. Kashyap [12] proposed SABS (Spatial Averaging Based Steganalysis) to detect stego videos. Differences of two averaging filters were calculated, and the same features as [3] were extracted for classification.

Despite of potentially high time-complexity, in order to obtain better detection performance on each frame, some image steganalytic methods are almost directly employed for video steganalysis. A framework considering video steganalysis as an extension of image steganalysis was proposed in [14], consisting of collusion, several video codec algorithms (e.g., motion estimation), and image steganalytic methods. In [28], the embedding operation was modeled as the convolution of the cover and the secret message on the histogram of adjacent frames' differences. Aliasing degree was then defined as the feature to detect the presence of hidden messages. Liu etc. [15] extracted Markov features from the differences of neighboring coefficients in the transform domain and achieved a satisfying result for the detection of MSU StegoVideo. Kancherla [10, 11] introduced the JPEG steganalytic features

[22] into spatial video steganalysis of MSU StegoVideo. Motion estimation was used to get an estimated copy of the cover object. L1 norm between features of the given object and the estimated object was employed to form the complete features.

In general, one kind of existing steganalytic methods based on TFA, PEF (Prediction Error Frame) or spatial average filtering extract features from the global statistical characteristics (like kurtosis, skewness, and 25 % percentile), while ignoring the correlation between either temporally or spatially neighboring pixels. The other kind of methods derived from image steganalytic methods are generally of high computing complexity and make insufficient use of the temporal redundancy. The most related works to this paper include Vinod's work in [18] which belongs to the former kind and Pevny's work in [20] which belongs to the latter one. Vinod suggested that lower dependencies within the video content exist in PEFs (Prediction Error Frames) than in the original video frames, and extracted features based on the distribution of the local variances of PEF samples. In pevny's method, no spatial redundancy was deployed. Differences of spatially neighboring pixels were modeled by a Markov chain and empirical probability transition matrices were calculated to form features, which were called the SPAM (Subtractive Pixel Adjacency Model) features. Since designed for image steganalysis, no temporal redundancy is utilized in Pevny's method.

This paper originates in the thought of combining the utilization of temporal redundancy and spatial redundancy. We focus on PEFs and model the differences of adjacent PEF samples by a Markov chain. Motion estimation makes use of the temporal redundancy between adjacent frames, while the differential filtering utilizes the spatial redundancy between adjacent PEF samples to further suppress the video content and amplify the stego noise. Those two kinds of processing lead to a higher WSNR (Watermark Signal to Noise Ratio), which seems favorable for steganalysis.

The paper is organized as follows. Section II starts with the rationale of proposed features: First, the model of spatial video steganography is stated. Second, the correlation between PEF and collusion is discussed theoretically, and comparison of PEF and PVD (Pixel Value Difference) is given according to the WSNR. At the last of this section, the proposed SPEAM (Subtractive Prediction Error Adjacency Model) features are presented, followed by a discussion of parameters in the feature extraction. The subsequent Section III presents the major part of experiments consisting of 1) comparison of several versions of the SPEAM features differing in the range of block matching, 2) comparison to SPAM and prior art of video steganalysis on uncompressed video sequences, and 3) compressed video sequences with format of MPEG2 and H.264. The conclusions are drawn in Section IV.

2 Proposed SPEAM features

Existing methods are mostly derived from collusion and image steganalysis. Aiming at amplifying the WSNR of the frame for analysis, collusion effectively suppresses the temporal redundancy between adjacent frames, while original image steganalytic methods which commonly derive features from PVDs satisfactorily suppress the spatial redundancy between adjacent pixels. In this section, the model of spatial video steganagraphy is stated, which is the basis for subsequent analysis. For illustrating why we derive our features from PEFs other than collusion or PVDs, the correlation between PEF and collusion is discussed, and the comparison of PEF and PVD is analyzed according to the WSNR. Finally, the proposed SPEAM features are presented.

2.1 Model of spatial video steganography

In a video steganographic system, the cover video is denoted by $U_k(m,n)$, where $k=1,2,\ldots K$ is the frame number, and $m=1,2,\ldots,M$ and $n=1,2,\ldots,N$ are row indices and column indices of pixels in each frame respectively. Before embedding, the secret message is modulated into a signal using a pseudo-random sequence, resulting in $W_k(m,n)$. As in [3], we call $W_k(m,n)$ the watermark. We assume that the embedding operation is employed in the spatial domain, and the related steganalysis is designed against spatial video steganography. Even if the embedding operation is carried out in a non-spatial domain such as the DCT (Discrete Cosine Transform) domain, and the DFT (Discrete Fourier Transform) domain, similar results can be formulated. The embedding operation is modeled as in [3] by

$$X_k(m,n) = U_k(m,n) + \alpha_k(m,n)W_k(m,n),\ k = 1,2,\ldots,K \tag{1}$$

where $\alpha_k(m,n)$ is a scaling factor used to tradeoff non-perceptibility and robustness. For simplicity of analysis, α is considered to be constant over all of the pixels and frames to give

$$X_k(m,n) = U_k(m,n) + \alpha W_k(m,n),\ k = 1,2,\ldots,K \tag{2}$$

Only the Y component of video frames is taken into consideration unless emphasized. For LSB steganography, we can get $W_k(m,n)=\pm 1$ and $\alpha=1$. For SSIS [16], $W_k(m,n)$ is treated as a 2-dimension Gaussian i.i.d. random process which obeys $N\left(0,\sigma_w^2\right)$. The scaled watermark $\alpha W_k(m,n)$ is a function of hidden messages, the scaling factor and the secret key.

2.2 Correlation between collusion and PEF

Spatial redundancy has been utilized for image steganalysis by some filtering methods such as the differential filter and the wavelet filter [21]. Spatial redundancy has also been used for video steganalysis by various methods such as spatial averaging [12, 23]. Besides spatial redundancy, there is still temporal redundancy in video sequences.

Collusion [3] is a classical method to employ temporal redundancy for steganalysis. When the watermark is embedded into the slow-moving video, collusion seems a good method to pre-process the stego video. On the contrary, if the cover video is fast-moving, motion estimation is needed as preprocessing before collusion.

The art of collusion calculates average values of pixels in the same position of adjacent frames, i.e., the average of $U_k(m,n)$, $U_{k-1}(m,n)$ and $U_{k+1}(m,n)$. Motion compensated collusion gains $\left(U_k(m,n)+\overline{U}_{k-1}(m,n)+\overline{U}_{k+1}(m,n)\right)/3$, where $\overline{U}_{k-1}(m,n)$ comes from the corresponding block of $U_k(m,n)$ in U_{k-1}. Those average values are then subtracted by $U_k(m,n)$ to get the residual signal, from which steganalytic features are extracted. Motion estimation is a high-complexity operation, which segments each frame into several blocks, and searches for the corresponding blocks of the current frame $U_k(m,n)$ in a specific region of its reference frame (e.g., U_{k-1}).

PEFs are got by replacing the averaging operation in collusion with the differential filtering $U_k(m,n)-\overline{U}_{k-1}(m,n)$. For compressed videos, PEFs can be gained directly from the compressed bit stream, while motion compensated collusion needs two copies of PEFs which the compressed bit stream of some codec (e.g., MPEG1) may not contain. This is why we would rather focus on PEFs other than collusion.

Figure 1 shows the joint probability $\Pr(U_k(m,n),U_{k-1}(m,n))$ and $\Pr(U_k(m,n),\overline{U}_{k-1}(m,n))$ estimated from 3710 frames of 14 standard video sequences (found at http://trace.eas.asu.edu/yuv/index.html. Each video sequence has not more than 300 frames)

captured with CIF size. Due to the high correlation between adjacent frames, the values of pixels in the same positions (or in the corresponding positions) of adjacent frames are close to each other. For fast-moving videos, the deviation between $U_k(m,n)$ and $\overline{U}_{k-1}(m,n)$ is much smaller than the deviation between $U_k(m,n)$ and $U_{k-1}(m,n)$, which suggests $U_k(m,n) - \overline{U}_{k-1}(m,n)$ used below may be less correlated to the video content than $U_k(m,n) - U_{k-1}(m,n)$. Figure 1 also suggests that the profile of the ridge along the major diagonal does not change much with the pixel value. This observation allows us to model the pixels in video frames by working with the differences $\mathrm{U}_k(m,n) - \overline{U}_{k-1}(m,n)$ instead of the co-occurrences $\left(\mathrm{U}_k(m,n), \overline{\mathrm{U}}_{k-1}(m,n)\right)$, which greatly reduces the model dimensionality. Further simplification can be achieved by only focusing on the differences falling in a certain range. If well set, this range may tradeoff the performance and complexity of the detector.

To observe the correlation between two adjacent frames more clearly, two assumptions are made as below:

1) There is always high correlation between $\mathrm{U}_k(m,n)$ and $\overline{U}_{k-1}(m,n)$ in the host video.
2) The watermark frames $\mathrm{W}_k(m,n)$ are independent to U_k, and are independent to each other. In addition, W_k obeys a Gaussian distribution $N\left(0, \alpha^2\sigma_w^2\right)$, where σ_w^2 denotes the variance of the stego noise and α denotes the embedding intensity.

The first assumption allows us to make use of the differences $U_k(m,n) - \overline{U}_{k-1}(m,n)$ falling in a small range. It is not satisfied when motion estimation is not precise enough (e.g., motion searching range is set too small; videos are quick-moving with irregular trail or taken with low frame rate). The second assumption leads to simpler analysis of SS steganography.

2.3 Comparison of PVD and PEF

Original PVDs before corrupted can be calculated by

$$\mathrm{PVD}_k(m,n) = \mathrm{U}_k(m,n) - \mathrm{U}_k(m,n+1) \tag{3}$$

and original PEFs can be got by

$$\mathrm{P}_k(m,n) = \mathrm{U}_k(m,n) - \overline{\mathrm{U}}_{k-1}(m,n) \tag{4}$$

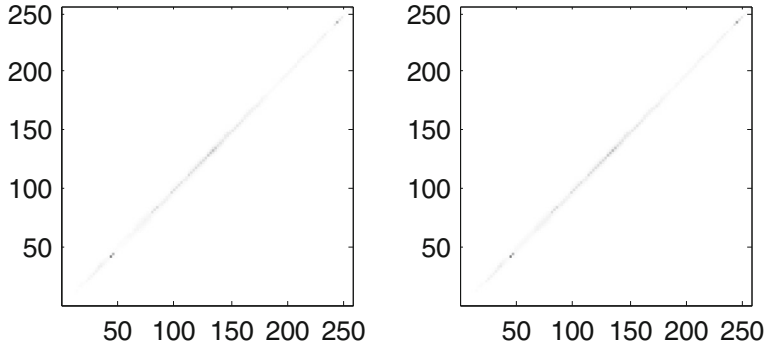


Fig. 1 Distribution of two pixels $(U_k(m,n), U_{k-1}(m,n))$ in the same position of adjacent frames (the left figure), and distribution of two pixels $\left(U_k(m,n), \overline{U}_{k-1}(m,n)\right)$ in the correlated positions of adjacent frames (the right figure) estimated from 3710 frames of 14 standard video sequences. The degree of gray at (x,y) in the figure is the probability of $\Pr(U_k(m,n) = x \wedge U_{k-1}(m,n) = y)$

After the embedding of SS steganography, PVDs are calculated by

$$\begin{aligned} \mathrm{PVD}_k^{'}(m,n) &= \mathrm{Y}_k(m,n) - \mathrm{Y}_k(m,n+1) \\ &= \mathrm{U}_k(m,n) - \mathrm{U}_k(m,n+1) \\ &+ \mathrm{W}_k(m,n) - \mathrm{W}_k(m,n+1) \end{aligned} \tag{5}$$

and PEFs are represented by the equation

$$\begin{aligned} \mathrm{P}_k^{'}(m,n) &= \mathrm{Y}_k(m,n) - \overline{\mathrm{Y}}_{k-1}(m,n) \\ &= \mathrm{U}_k(m,n) - \overline{\mathrm{U}}_{k-1}(m,n) \\ &+ \mathrm{W}_k(m,n) - \overline{\mathrm{W}}_{k-1}(m,n) \end{aligned} \tag{6}$$

The distribution of $\mathrm{W}_k(m,n) - \overline{\mathrm{W}}_{k-1}(m,n)$ is the same as that of $\mathrm{W}_k(m,n) - \mathrm{W}_k(m,n+1)$ according to the second assumption, which allows us to just focus on the remaining components of PVDs (i.e., $\mathrm{U}_k(m,n) - \mathrm{U}_k(m,n+1)$) and PEFs (i.e., $\mathrm{U}_k(m,n) - \overline{\mathrm{U}}_{k-1}(m,n)$). It is quite difficult to compare those two components using existing mathematical models, and thus, we focus on the first moment and second moment of PVDs and PEFs got from the cover videos according to the experiments and analysis below.

Each frame in the original videos is segmented into several blocks of size 8×8 for block matching. Local variances of PVDs and PEFs with size of 3×3 are denoted by $Var_{\mathrm{PVD}_{k,i}}$ and $Var_{\mathrm{P}_{k,i}}$, where i is the indices of blocks in a frame. The probabilities of three cases of $\left(Var_{\mathrm{PVD}_{k,i}}, Var_{\mathrm{P}_{k,i}}\right)$ from all the original video frames are shown in Table 1.

It is interesting that in most blocks of the original videos, $Var_{\mathrm{P}} < Var_{\mathrm{PVD}}$ occurs whether the video content is fast-moving or not. On the other hand, when motion estimation is perfect as we expect, we can get $E[\mathrm{P}_k(m,n)] \approx 0$ for all the cover video

Table 1 Probabilities of three cases of $\left(Var_{\mathrm{PVD}_{k,i}}, Var_{\mathrm{P}_{k,i}}\right)$, where P {>} implies the probability of $P\{Var_{\mathrm{P}_{k,i}} > Var_{\mathrm{PVD}_{k,i}}\}$

ID	Name	Frames	Camera motion	Object motion	P {>}	P {=}	P {<}
1	Akiyo	300	N/A	slow	0.010	0	0.990
2	Bus	150	panning	global(fast)	0.593	0	0.407
3	Coastguard	300	panning	Translational	0.222	0	0.778
4	Container	300	N/A	Slow	0.074	0	0.926
5	Flower	250	translational(fast)	N/A	0.138	0.000	0.862
6	Hall	300	N/A	non-translational	0.260	0.011	0.730
7	Highway	300	non-translational(fast)	non-translational(fast)	0.316	0.059	0.625
8	Mobile	300	panning	translational, rotational	0.057	0	0.943
9	Mother-daughter	300	N/A	local(fast)	0.117	0.022	0.860
10	News	300	N/A	local(fast)	0.061	0	0.939
11	Silent	300	N/A	local(fast)	0.107	0	0.893
12	Stefan	90	non-translational(fast)	global(fast)	0.407	0	0.593
13	Tempete	260	zooming	slow	0.180	0	0.820
14	Waterfall	260	zooming	slow	0.006	0	0.994
Average					0.1548	0.0075	0.8736

frames, while $E[\mathrm{PVD}_k(\mathrm{m},\mathrm{n})] \geq 0$ is inherent for PVDs. A simple comparison of the WSNR of PVD and PEF for the case that motion estimation is ideally accurate is given by

$$\begin{aligned} WSNR_{\mathrm{PVD}_k} &= \frac{E\left[(\mathrm{W}_k(m,n)-\mathrm{W}_k(m,n+1))^2\right]}{E\left[(\mathrm{U}_k(m,n)-\mathrm{U}_k(m,n+1))^2\right]} \\ &= \frac{2\alpha^2\sigma_w^2}{Var_{\mathrm{PVD}_k}+E^2[\mathrm{PVD}_k(m,n)]} \\ &\leq \frac{2\alpha^2\sigma_w^2}{Var_{\mathrm{PVD}_k}} \leq \frac{2\alpha^2\sigma_w^2}{Var_{\mathrm{P}_k}} \\ &\approx \frac{E\left[\left(\mathrm{W}_k(m,n)-\overline{\mathrm{W}}_{k-1}(m,n)\right)^2\right]}{E\left[\left(\mathrm{U}_k(m,n)-\overline{\mathrm{U}}_{k-1}(m,n)\right)^2\right]} = WSNR_{\mathrm{PEF}_k} \end{aligned} \tag{7}$$

A larger WSNR in PEFs based on ideally motion estimation than in PVDs suggests that features based on PEFs may be more efficient than those based on PVDs.

2.4 The SPEAM features

As mentioned above, PEF denoted by $\mathrm{P}_k(m,n) = \mathrm{U}_k(m,n) - \overline{\mathrm{U}}_{k-1}(m,n)$ seems to be a favorable variable for steganalysis. Figure 2 shows $\Pr(\mathrm{P}_k(m,n), \mathrm{P}_k(m,n+1))$ of the original video sequence "akiyo", "akiyo" corrupted with α=1, and the original video sequence "waterfall". It is quite easy to distinguish the first two cases. However, there seems no obvious deviation between the corrupted "akiyo" and uncorrupted "waterfall", which suggests $\Pr(\mathrm{P}_k(m,n),\ \mathrm{P}_k(m,n+1))$ may be still correlated with the content of videos.

In fact, motion estimation to obtain PEFs has suppressed temporal redundancy in the video content, while spatial redundancy inherited from frame samples still exists within PEF samples. We employ an additional differential filter to realize further suppression of spatial redundancy within $(\mathbf{P}_k(m,n+1), \mathbf{P}_k(m,n))$. The differential filter is denoted by

$$\mathrm{D}_k^{\rightarrow}(m,n) = \mathrm{P}_k(m,n) - \mathrm{P}_k(m,n+1) \tag{8}$$

where $\mathrm{D}_{\mathrm{k}}^{\rightarrow}(\mathrm{m},\mathrm{n})$ denotes the left-to-right difference of PEF samples. In addition, instead of the joint probability $\Pr(\mathrm{D}_k(m,n+1),\ \mathrm{D}_k(m,n))$, a more commonly used conditional probability $\Pr(\mathrm{D}_k(m,n+1)/\mathrm{D}_k(m,n))$ is calculated to model correlations between adjacent PEF samples.

Figure 3 summarizes the feature extraction process of the SPEAM features, where the SPEAM implies modeling of adjacent PE samples' differences by a Markov chain. First, difference matrices of adjacent PEF samples are computed. Second, transition probabilities of difference matrices along the same direction are calculated. Finally, several subsets of those transition probability matrices are averaged into two Markov matrices, which form the SPEAM features.

The Markov chain is chosen here mainly because of two facts. The first fact is that Markov features have performed well for image steganalysis, which implies that the Markov chain is

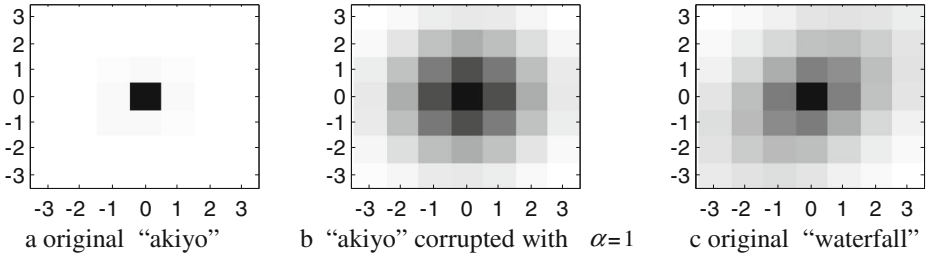


Fig. 2 $\Pr(\mathrm{P}_k(m,n), \mathrm{P}_k(m,n+1))$ of original video sequence "akiyo", "akiyo" corrupted with α=1, and original video sequence "waterfall", **a** original "akiyo", **b** "akiyo" corrupted with α=1, **c** original "waterfall"

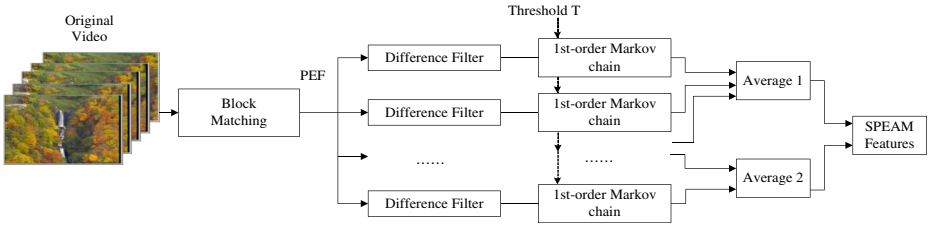


Fig. 3 Scheme of extraction of SPEAM features

useful for modeling spatially adjacent pixels and is favorable for steganalysis. The other fact is that adjacent PEF samples in the same PEF are quite similar to adjacent pixels in the same image. To avoid rigorous analysis of the complex dependencies between adjacent PEF samples theoretically, we attempt to introduce the Markov chain to model the dependencies. Since dependencies between adjacent PEF samples are manipulated when the secret message is embedded into the cover, we extract Markov features for detecting the existence of the secret.

The steps of feature extraction are as follows.

Step1: Calculate difference matrices.

Difference matrices are denoted by $\mathrm{D}^{\bullet}_{k\,(m,n)}$, where $k\in\{1,\ldots,K\}$ represents the frame indices, $\bullet\in\{\leftarrow,\rightarrow,\uparrow,\downarrow,\nwarrow,\searrow,\nearrow,\swarrow\}$ gives the direction of difference. For $\bullet='\uparrow'$

$$\mathrm{D}^{\uparrow}_{k}(m,n)=\mathrm{P}_{k}(m,n)-\mathrm{P}_{k}(m-1,n) \tag{9}$$

where m, n are the row and column indices. For $\bullet='\nearrow'$

$$\mathrm{D}^{\nearrow}_{k}(m,n)=\mathrm{P}_{k}(m,n)-\mathrm{P}_{k}(m-1,n+1) \tag{10}$$

Other difference matrices are obtained in similar manners.

Step2: Compute transition probabilities of difference matrices along the same direction.

The SPEAM features model difference matrices $\mathrm{D}^{\bullet}_{k}$ by a first-order Markov process and compute the empirical matrices. For $\bullet='\uparrow'$, the empirical matrix is given by

$$\mathrm{M}^{\uparrow}_{k,u,v}=\frac{1}{MN}\sum_{n}\sum_{m}\Pr\left(\mathrm{D}^{\uparrow}_{k}(m-1,n)=v|\mathrm{D}^{\uparrow}_{k}(m,n)=u\right) \tag{11}$$

where u, $v\in\{-T,\ldots T\}$. T denotes the threshold of u,v we concern. If $\Pr\left(\mathrm{D}^{\uparrow}_{k,m,n}=v\right)=0$, then $\mathrm{M}_{k,u,v=0}$. For $\bullet='\nearrow'$

$$\mathrm{M}^{\nearrow}_{k,u,v}=\frac{1}{MN}\sum_{n}\sum_{m}\Pr\left(\mathrm{D}^{\nearrow}_{k}(m-1,n+1)=v|\mathrm{D}^{\nearrow}_{k}(m,n)=u\right) \tag{12}$$

It should be noted that the differential directions of the two matrices $\mathrm{D}^{\bullet}_{k}$ in Eq. (11) (or (12)) are the same. Other empirical matrices are obtained in similar manners.

Step3: Average Markov matrices to get two final matrices.

To decrease the feature dimensionality, we simply average the matrices $\mathrm{M}^{\bullet}_{k,u,v}$ with the same distances between the two difference matrices in the calculations of $\mathrm{M}^{\bullet}_{k,u,v}$ (e.g., $\mathrm{D}^{\uparrow}_{k}\times(m,n)=u$ and $\mathrm{D}^{\uparrow}_{k}(m-1,n)$ for $\mathrm{M}^{\uparrow}_{k,u,v}$ in (11)). For $\bullet\in\{\leftarrow,\rightarrow,\uparrow,\downarrow\}$, the distance is thought to be the same one, while for $\bullet\in\{\nwarrow,\searrow,\nearrow,\swarrow\}$, the distance is assigned to be the

other one. According to the two distinct distances, the matrices $\mathrm{M}^{\bullet}_{k,u,v}$ of all the 8 directions are separated into two subsets, which are then averaged respectively. With a slight abuse of notation, the SPEAM features of a PEF can be formally written as

$$\mathrm{F}^{k}_{1,\ldots,m} = \frac{1}{4}\left(\mathrm{M}^{\leftarrow}_{k} + \mathrm{M}^{\rightarrow}_{k} + \mathrm{M}^{\uparrow}_{k} + \mathrm{M}^{\downarrow}_{k}\right) \tag{13}$$

$$\mathrm{F}^{k}_{m+1,\ldots,2m} = \frac{1}{4}\left(\mathrm{M}^{\nwarrow}_{k} + \mathrm{M}^{\searrow}_{k} + \mathrm{M}^{\nearrow}_{k} + \mathrm{M}^{\swarrow}_{k}\right) \tag{14}$$

where the dimensionalities of $\mathrm{M}^{\bullet}_{k}$, $\mathrm{F}^{k}_{1,\ldots,m}$ and $\mathrm{F}^{k}_{m+1,\ldots,2m}$ are the same, i.e., $m = (2T+1)^2$.

There are two main parameters in the extraction of SPEAM features. One parameter is the searching range in the motion estimation scheme. A larger searching range may bring a higher motion estimation precision, but takes more time. Experiments of the searching range are presented in Section III.A. The other parameter is the upper bound of $\left|\mathrm{D}^{\bullet}_{k}\right|$, which is represented by T. A larger T implies taking more cases of adjacent PE samples' differences into consideration. When T is too large, however, most cases of adjacent PE samples may be unrelated to steganography and may be useless for steganalysis.

In [20], Pevny has employed T=4 as the upper bound of difference values of adjacent pixels in image steganalysis, and experiments have proved its usefulness. Here we calculate the matrix $\mathrm{F}^{k}_{1,\ldots,m}$ of all the 3710 frames from 14 standard video sequences, and intend to decide T from the average of $\mathrm{F}^{k}_{1,\ldots,m}$. Figure 4 gives the average $\mathrm{F}^{k}_{1,\ldots,m}$ of all the 3710 original video frames, stego frames embedded with SS of α=1, and stego frames embedded with SS of α=3. Three figures at the top are amplified to obtain the bottom figures. It should be noted that the blank samples lying near the anti-diagonal line are mainly caused by i.i.d. random numbers.

Figure 4 implies that based on the range $[-3,3]\times[-3,3]$ of $\mathbf{F}^{k}_{1,\ldots,m}$, we can manually distinguish the stego videos from the cover ones. Our tests in the next section have also shown that it is effective to set T=3, leading to the features' dimension of $2 \bullet (2T+1)^2 = 98$.

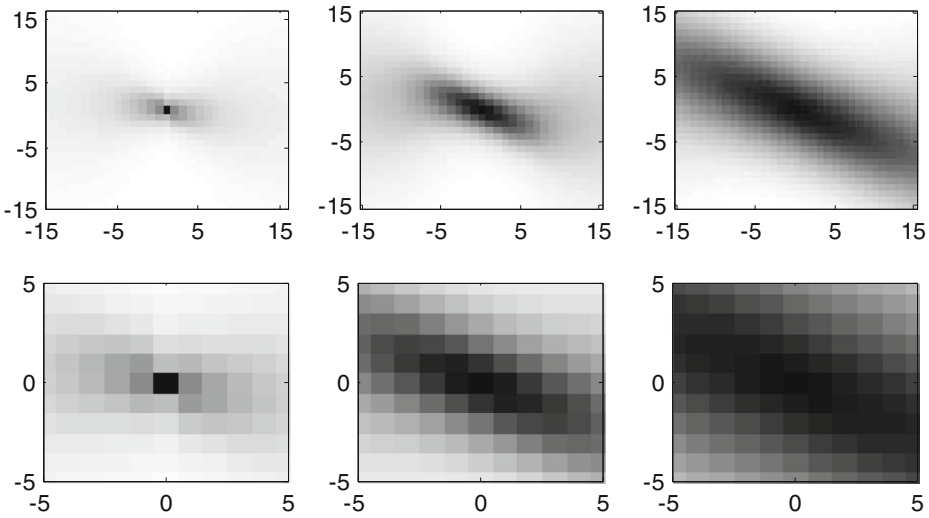


Fig. 4 $\mathrm{F}^{k}_{1,\ldots,m}$ of all the original video frames (left), stego frames embedded with SS of α=1 (*middle*), and stego frames with SS of α=3 (*right*). The upper figures are amplified to form the lower ones

3 Steganalysis of spread spectrum steganography using SPEAM features

To evaluate the performance of the SPEAM features, we test them against SS steganography which is a broadly-used embedding method in video spatial steganography. SS methods can be simply categorized into two kinds [18] when used in watermarking. The first kind embeds the same watermark pattern in all video frames, while the other kind never embeds the same watermark pattern in two distinct frames. We mainly care about the latter, which is more close to actual steganography.

Standard video sequences are usually used for researches on video steganalysis, video codec, object tracking etc. Video sequences found at http://trace.eas.asu.edu/yuv/index.html are used here. The size of those video frames is CIF (352×288), and the frame rate is 30fps. For simplicity, only the first 90 frames of each video are used here for the experiments. Contents of them are listed in Table 1. SVM [5] is used as the classifier, and the radial basis function kernel is employed to implement the transformation of the feature vector for SVM. Grid searching is exploited to find the optimal parameter pair (C,γ), where C is the penalty parameter, and γ is the controlling parameter of the kernel function. All grid points of (C={1e2,1e3,1e4},γ=−$\log_2$(98)+{−3,−2,…,4}) are tested.

Each video sequence has a single scene, leading to high dependencies between distinct frames which are even not neighboring. This makes it unreasonable to divide each sequence into several sub-sequences and take experiments upon those sub-sequences. Sequence-level cross validation stated in Algorithm 1 is designed to evaluate the proposed features. The accuracy Acc(C_o,γ_o) correlated to the optimal parameters (C_o,γ_o) forms the final testing result. The accuracy of each loop in Algorithm 1 is calculated by

$$\mathrm{Acc\_iter}(i) = \frac{TP + TN}{N} \tag{15}$$

where TP is true positive, and TN is true negative. When 5 videos are chosen for testing, the whole loops of i for each (C,γ) is as large as $C_{14}^{5} = 2002$. For simplicity, we set the maximum of i to be 100.

Algorithm 1 Sequence-level Cross Validation

Input: Features of each video tested, the number of videos, and all the grid points of parameters $\{C,\gamma\}$

Output: The optimal parameters (C_o,γ_o), and the accuracy of the classification $\mathbf{Acc}(C,\gamma)$.

```
foreach C in {C,γ}
  foreach γ in {C,γ}
    set Acc_iter to 0;
    foreach i from 1 to 100
        randomly choose the features of 5 videos for testing;
        choose the features of other 9 videos for training;
        train and test using SVM with C,γ, and record the accuracy Acc_iter(i);
    end
    calculate the average accuracy Acc(C,γ) = 1/100 Σ_{i=1}^{100} Acc_iter(i);
  end
end
output (C_o,γ_o) = arg max(Acc(C,γ)) and Acc.
```

3.1 Motion searching range

The SPEAM features with searching range of 0, 3, and 7 in the motion estimation scheme are tested. Corresponding results are given in Table 2. The SPEAM features, which are calculated with the motion searching range of • are denoted by SPEAM(•).

Corrupted pixel ratio (cpr), which is similar to bits per pixel (bpp) commonly used in image steganography, is defined here to represent the ratio of the corrupted pixel number to the total pixel number.

Generally speaking, a larger motion searching range implies a more precise matching of blocks in two adjacent frames. This makes it easier for the steganalyzer to distinguish the cover from the stego objects. Table 2 has shown that in most cases, the larger the searching range is set, the better results we obtain. However, deviations between SPEAM(3) and SPEAM(7) are not obvious. The reason may be that precision of motion estimation depends on not only the searching range of block matching, but also the content of videos, the size of blocks, the precision of motion unit (such as pixel, sub-pixel, and quarter pixel), and effects of the embedding operation, etc.

As a larger searching range takes more time for the feature extraction scheme of uncompressed videos, we choose 3 for the tradeoff of motion estimation precision and complexity in the following experiments of uncompressed video sequences.

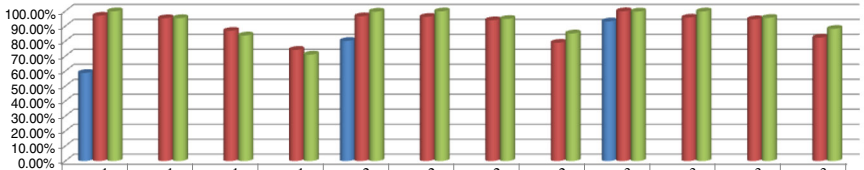
3.2 Spatial steganalysis of uncompressed video sequences

To compare the SPEAM features with the SPAM features and Budhia's features in [3] (Block-based collusion features are tested here with motion searching range of 3), experiments using sequence-level cross validation are taken here on uncompressed video sequences. Figure 5 gives the detection accuracy. Because of the poor performance of Budhia's features, we just test them for cpr=1.

Results shown in Fig. 5 imply that for all the cases tested, the SPEAM and SPAM features are close and much better than Budhia's features. This may be because more characteristics (i.e., the Markov features) of the video sequences are utilized for steganalysis in the SPAM and SPEAM features. The time consumed by SPEAM, however, is more than that of

Table 2 Detection accuracy of the SPEAM features with motion searching range of 0, 3, and 7 on uncompressed videos

α	cpr	SPEAM(0)	SPEAM(3)	SPEAM(7)
1	1.00	98.16%	99.90%	99.98%
1	0.50	91.20%	95.37%	96.48%
1	0.25	82.76%	83.72%	83.48%
1	0.10	70.66%	70.99%	71.91%
2	1.00	98.23%	99.64%	99.78%
2	0.50	98.34%	99.84%	99.84%
2	0.25	92.80%	94.79%	95.92%
2	0.10	84.15%	85.08%	87.16%
3	1.00	97.44%	99.74%	99.76%
3	0.50	97.51%	99.81%	99.37%
3	0.25	96.27%	95.53%	96.29%
3	0.10	85.72%	88.17%	89.23%



	α=1 cpr=1	α=1 cpr=0.5	α=1 cpr=0.25	α=1 cpr=0.1	α=2 cpr=1	α=2 cpr=0.5	α=2 cpr=0.25	α=2 cpr=0.1	α=3 cpr=1	α=3 cpr=0.5	α=3 cpr=0.25	α=3 cpr=0.1
Budhia's feature	58.77%				80.11%				93.13%			
SPAM	96.93%	95.27%	86.89%	74.17%	96.55%	96.19%	93.91%	78.85%	99.95%	95.70%	94.59%	82.26%
SPEAM	99.90%	95.37%	83.72%	70.99%	99.64%	99.84%	94.79%	85.08%	99.74%	99.81%	95.53%	88.17%

Fig. 5 Detection accuracy of the SPAM features, the SPEAM(3) features and Budhia's features on uncompressed video sequences

Budhia's features, which is due to the calculation of probability transition matrices (such as if-else decision) and block matching.

Figure 5 also implies that SPAM and SPEAM are generally similar for the spatial steganalysis of uncompressed videos. This may be because that for uncompressed videos, spatial dependencies between adjacent pixels may contain much enough information to distinguish the stego videos from the cover videos, while combining the utilization of temporal and spatial redundancy does not bring additional information to obviously improve the detection accuracy.

With a more precise block matching scheme which gets a matching result from the stego object more close to that of the cover one, the SPEAM features are believed to be more favorable.

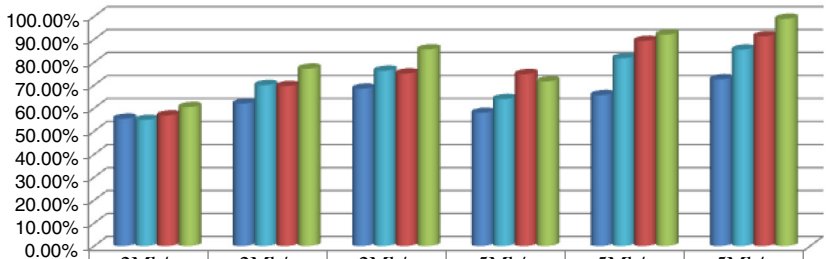
3.3 Spatial steganalysis of compressed video sequences(MPEG2)

To further evaluate the performances of the proposed features for actual steganalytic systems, the experiment in the last subsection is replicated here on compressed video sequences. SS steganography is implemented on cover videos of YUV format, which are then converted to MPEG2 format by VcDemo [26] with GOP structure of IBBPBBPBBPBB and motion searching range of 15. At last, features are extracted from the cover MPEG2 videos and the stego MPEG2 videos, and sequence-level cross validation is employed to obtain testing results.

Since PEFs can be obtained after partial decompression of the compressed videos, no additional block matching is needed for the extraction of the SPEAM features. The Markov features of PEFs are calculated directly to form the SPEAM features. For a B-type frame, features of two PEFs are averaged, while for a P-type frame, features of the only PEF are directly employed.

For simplicity, we just use SS for message embedding, regardless of whether the message in the MPEG2 videos can be completely extracted. As the compressing scheme may erase some of the watermarks, only cpr=1 is tested here.

Figure 6 shows the detection accuracy of the SPEAM features, the SPAM features, Budhia's features, and Vinod's features [18]. When the bit rate is as low as 2 Mb/s, the distortion of watermark and video content leads to degradation of all the tested features' performances. Contrarily, when the bit rate is 5 Mb/s, the simulation result is similar as result shown in Table 2. This is because the quality of videos which have bit rate of 5 Mb/s is close to that of uncompressed videos (The bit rate of uncompressed videos is $352\times288\times30\approx$ 3.04 Mb/s).



	2Mb/s α=1	2Mb/s α=2	2Mb/s α=3	5Mb/s α=1	5Mb/s α=2	5Mb/s α=3
Budhia's features	55.34%	61.91%	68.44%	57.89%	65.63%	72.53%
Vinod's features	54.83%	69.99%	76.25%	63.92%	81.70%	85.49%
SPAM	56.76%	69.58%	75.05%	74.77%	89.29%	91.27%
SPEAM	60.46%	77.17%	85.56%	71.62%	92.02%	98.92%

Fig. 6 Detection accuracy of Budhia's features, Vinod's features, the SPAM features, and the SPEAM features on compressed video sequences (MPEG2). Bitrates of 2 Mb/s and 5 Mb/s, and α of 1, 2, 3 are tested

In all the cases tested, the SPEAM features and the SPAM features perform much better than other two features, which suggests that modeling with a Markov chain contains more information sensitive to steganography than the i.i.d. model used in Budhia's features and Vinod's features. Besides, the SPEAM features seem prior to the SPAM features in most cases. The detailed ROC curves of the SPAM features and the SPEAM features are provided in Section III.E.

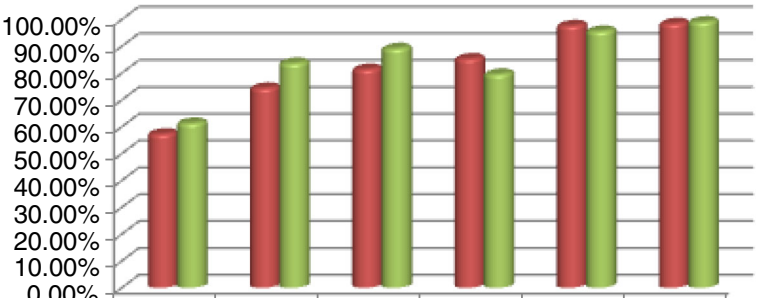
3.4 Spatial steganalysis of compressed video sequences(H.264)

Experiments in this subsection are carried out on compressed video sequences compressed in H.264 by libx264 encoder of FFMPEG [7]. Firstly, SS steganography is implemented on the cover videos. Secondly, both the cover videos and the stego videos are encoded into H.264 format by FFMEPG. Lastly, SPEAM features of videos are extracted and tested. The profile of the H.264 encoder is set to "baseline", and two cases of bit rate (i.e., 2 Mb/s. and 5 Mb/s) are tested.

Figure 7 shows the testing results of the SPEAM features and the SPAM features. Generally, the results shown in the figure are similar to testing results on videos compressed in MPEG2 format. This means when the bit rate is 5 Mb/s, two features perform similarly, while when the bit rate is 2 Mb/s, the SPEAM features are prior to the SPAM features. The detailed ROC curves are provided in the next subsection.

3.5 Experimental results of SPEAM and SPAM

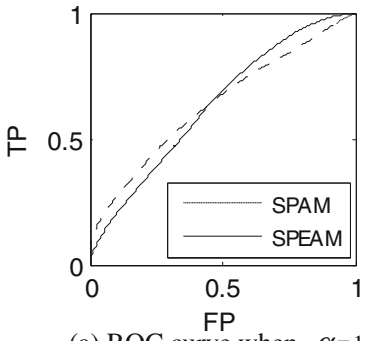
To compare the SPEAM features and the SPAM features for compressed videos, Fig. 8 gives the ROC curve of testing results when the SPAM features and the SPEAM features are tested on MPEG2-format videos, and Fig. 9 gives the ROC curve of testing results on H.264-format videos. Bit rate of 2 Mb/s and α of 1, 2, 3 are tested. Those two figures suggest that when $\alpha=1$, since most of the stego noise has been erased by the compression scheme, both features perform poor. When $\alpha=1.3$, distortion exists in the video content, and most of the stego noise survives the compression scheme. In this case, the SPEAM features which consider both spatial redundancy and temporal redundancy perform better than the SPAM



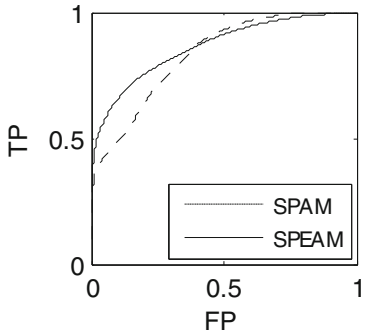
	2Mb/s α=1	2Mb/s α=2	2Mb/s α=3	5Mb/s α=1	5Mb/s α=2	5Mb/s α=3
SPAM	57.33%	74.41%	81.46%	85.37%	97.71%	98.46%
SPEAM	61.35%	83.71%	89.18%	79.64%	95.65%	99.10%

Fig. 7 Detection accuracy of the SPAM features and the SPEAM features on compressed video sequences (H.264). Bitrates of 2 Mb/s and 5 Mb/s, and α of 1, 2, 3 are tested

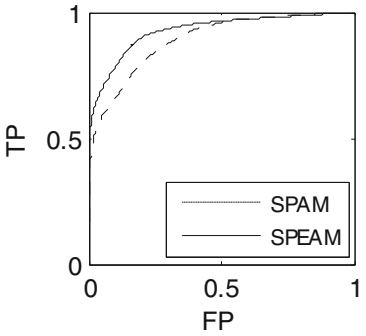
features which just consider spatial redundancy between adjacent pixels. This is why we believe the SPEAM features are favorable for spatial video steganalysis.



(a) ROC curve when $\alpha = 1$

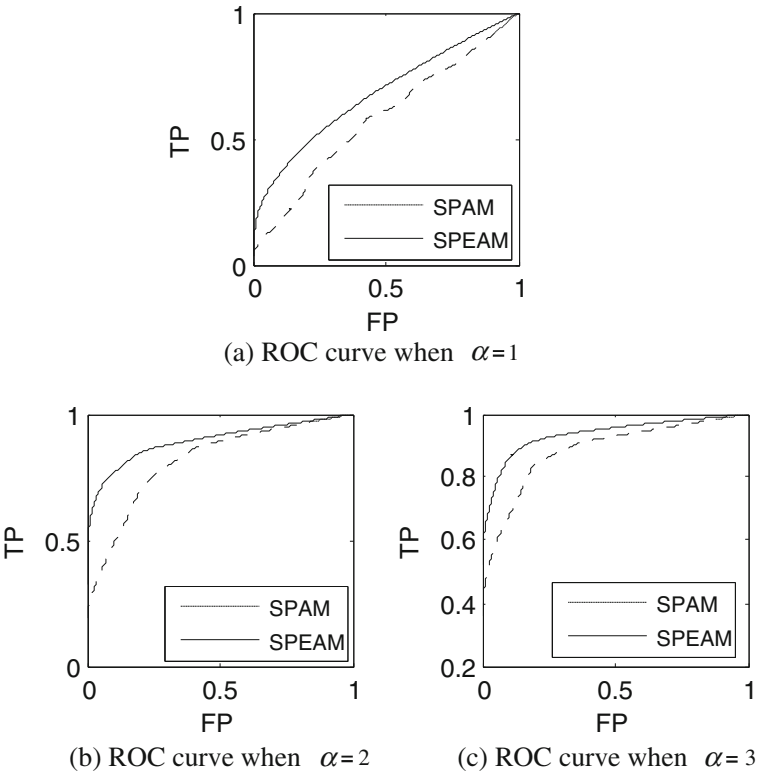


(b) ROC curve when $\alpha = 2$



(c) ROC curve when $\alpha = 3$

Fig. 8 ROC curve of the SPAM features and the SPEAM features on compressed video sequences (MPEG2). The bitrate of 2 Mb/s, and α of 1, 2, 3 are tested. **a** ROC curve when α=1, **b** ROC curve when α=2, **c** ROC curve when α=3



(a) ROC curve when $\alpha = 1$

(b) ROC curve when $\alpha = 2$

(c) ROC curve when $\alpha = 3$

Fig. 9 ROC curve of the SPAM features and the SPEAM features on compressed video sequences (H.264). The bitrate of 2 Mb/s, and α of 1, 2, 3 are tested. **a** ROC curve when α=1, **b** ROC curve when α=2, **c** ROC curve when α=3

4 Conclusions

The work presented in this paper utilizes the fact that the correlation between adjacent PEF samples exists in typical digital media while the dependences degrade because of the random stego noise. The dependences between differences of neighboring PEF samples are modeled by a Markov chain. Subsets of the empirical probability transition matrices are taken as a feature vector for steganalysis, which is called the SPEAM features.

The main advantage of SPEAM is that for compressed video sequences which are the major components of the Internet videos, SPEAM performs better than other methods tested. Furthermore, the calculation of features is of low complexity and is suitable for real-time applications. For uncompressed video sequences, SPEAM performs similar to SPAM which is one of the most effective image steganalytic methods, and is prior to previous works by Budhia.

In the future, we would like to investigate more advanced measures to merge the utilization of temporal redundancy and spatial redundancy, aiming at achieving better performances, especially for compressed videos with contents which are fast-moving with irregular trails or of high texture complexity. In addition, the effectiveness of the steganalytic features on videos of various codec should be further tested. Besides, steganography utilizing information got in the compression schemes (such as motion vector [1]) has been studied, and several steganalytic methods have been proposed [4, 25]. We also plan to

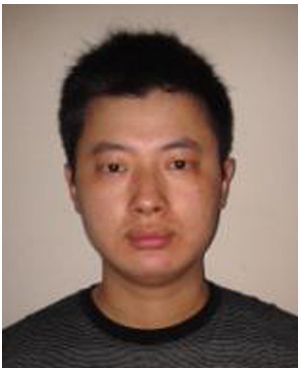
research on dependencies between intra-frame MVs and correlation within inter-frame MVs, and derive favorable features for steganalysis.

Acknowledgments This research was supported by the National Natural Science Foundation of China (NSFC) under the grant No. 61170226, the Fundamental Research Funds for the Central Universities under the grant Nos. SWJTU11CX047, and Chengdu Science and Technology program under the grant No. 12DXYB214JH-002.

References

1. Aly H (2011) Data hiding in motion vectors of compressed video based on their associated prediction error. IEEE Trans Inf Forensic Secur 6(1):14–18
2. Budhia U, Kundur D (April, 2004) "Digital video steganalysis exploiting collusion sensitivity", "Sensors, Command. Control, Communications, and Intelligence(C3I) Technologies for Homeland Security and Homeland Defense", Edward M. Carapezza, ed., Proc. SPIE, vol. 5403
3. Budhia U, Kundur D, Zourntos T (2006) Digital video steganalysis exploiting statistical visibility in the temporal domain. IEEE Trans Inf Forensic Secur 1(1):43–55
4. Cao Y, Zhao X, Feng D (2012) Video steganalysis exploiting motion vector reversion-based features. IEEE Signal Process Lett 19(1):35–38
5. Chang CC and Lin CJ "LIBSVM: a library for support vector machines"[Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
6. Cox J, Kilian J, Leighton T, Shamoon T (1997) Secure spread spectrum Watermarking for Multimedia. IEEE Trans Image Process 6(12):1673–1687
7. FFMPEG Library[Online], Available: http://ffmpeg.sourceforge.net/
8. Hartung F, Girod B (May 1998) "Watermarking of uncompressed and compressed video", Signal Processing, Special Issue on Copyright Protection and Access Control for Multimedia Services,66(3):283–301
9. Jainsky JS, Kundur D, Halverson DR (September 2007) "Towards Digital Video Steganalysis using Asymptotic Memoryless Detection", Proc. ACM MM&Sec'07, Dallas, Texas, USA, pp. 161–168
10. Kanchela K, Mukkamala S (June, 2009) "Video Steganalysis using Motion Estimation", Proc. of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, pp. 1510–1515
11. Kancherla K, Mukkamala S (2009) "Video Steganalysis Using Spatial and Temporal Redundancies", Proc. International Conference on High Performance Computing&Simulation, pp. 200–207
12. Kashyap S, Bora PK (2010) "Spatial Averaging based Steganalysis Scheme to detect Antipodal Watermarks", pp. 1–5
13. Li Q, Cox IJ (2007) Using perceptual models to improve fidelity and provide resistance to valumetric scaling for quantization index modulation watermarking. IEEE Trans Inf Forensic Secur 2(2):127–139
14. Liu B, Liu F, Yang CF (2008) "Stepwise inter-frame correlation-based steganalysis system for video streams", Security and Communication Networks, Security Comm. Networks, pp. 487–494
15. Liu Q, Sung AH, Qiao M (2008) "Video steganalysis based on the expanded Markov and joint distribution on the transform domains—Detecting MSU StegoVideo", in Proc. 7th International Conference on Machine Learning, and Applications, pp. 671–674
16. Marvel LM, Boncelet CG, Retter CT (1999) Spread spectrum image steganography. IEEE Trans Image Process 8(8):1075–1083
17. MSUStegovideo[Online], Available: http://compression.ru/video/stego_video/index_en.html.
18. Pankajakshan V, Doërr G, Bora PK (2009) Detection of motion-incoherent components in video streams. IEEE Trans Inf Forensic Secur 4(1):49–58
19. Pérez-González F, Barni M, Abrardo A, Mosquera C (2004) "Rational Dither Modulation: A Novel Data-Hiding Method Robust to Value-metric Scaling Attacks", IEEE 6th Workshop on Multimedia Signal Processing, pp. 139–142
20. Pevný T, Bas P, Fridrich J (2010) Steganalysis by subtractive pixel adjacency matrix. IEEE Trans Inf Forensic Secur 5(2):215–224
21. Pevný T, Bas P, Fridrich J (September, 2009) "Steganalysis by subtractive pixel adjacency matrix", in Proc. 11th ACM Multimedia &Security Workshop, Princeton, NJ, pp. 75–84
22. Pevný T, Fridrich J (February, 2007) "Merging Markov and DCT Features for Multi-Class JPEG Steganalysis", in Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, 6505:301–314

23. Rana V, Mishra R, Bora PK, Kashyap S (2008) "Novel Scheme of Video Steganalysis for Detecting Antipodal Watermarks", Proc. IEEE Region 10 Conference-TENCON, pp. 1–5
24. Sharp T (2001) "An implementation of key-based digital signal Steganography", in: Proc. 4th Information Hiding Workshop. Lect Notes Comput Sci 2137:13–26
25. Su Y, Zhang C, Zhang C (2011) A video steganalytic algorithm against motion-vector-based steganography. Signal Process 91(8):1901–1909
26. VCDemo: Image and Video Compression Learning Tool. [Online]. Available: http://ict.ewi.tudelft.nl/~inald/vcdemo
27. Zhang T, Li W, Zhang Y, Zheng E, Ping X (2010) Steganalysis of LSB matching based on statistical modeling of pixel difference distributions. Inf Sci 180:4685–4694
28. Zhang C, Su Y, Zhang C (2008) Video steganalysis based on aliasing detection. Elec Lett 44(13):801–803



Keren Wang received the M.S. degree from Zhengzhou Information Science and Technology Institute, China in 2010 and is now a candidate for doctor's degree on communication and information system at Zhengzhou Information Science and Technology Institute. His research interests include video steganography and steganalysis, motion estimation and applications, image and video classification.

Jiesi Han received the doctor's degree from Zhengzhou Information Science and Technology Institute, China in 2010. His research interests include steganography, steganalysis for image and video. He is a member of the China Computer Federation (CCF).

Hongxia Wang received the B.S. degree from Hebei Normal University, Shijiazhuang, in 1996, and the M.S. and Ph.D. degrees from University of Electronic Science and Technology of China, Chengdu, in 1999 and 2002, respectively. She engaged in postdoctoral research work in Shanghai Jiaotong University from 2002 to 2004. Currently she is a professor with School of Information Science and Technology, Southwest Jiaotong University, Chengdu. Her research interests include multimedia information security, digital forensics, and information hiding. She has published 50 peer research papers and wined 8 authorized patents.