http://doi.org/10.1016/j.diin.2013.10.004

Detection of Frame Deletion for Digital Video Forensics

Tamer Shanableh
Department of Computer Science and Engineering
College of Engineering
American University of Sharjah
Fax: +971 6 515-2979
tshanableh@aus.edu

Abstract

The abundance of digital video forms a potential piece of evidence in courtrooms. Augmenting subjective assessment of digital video evidence by an automated objective assessment helps increase the accuracy of deciding whether or not to admit the digital video as legal evidence. This paper examines the authenticity of digital video evidence and in particular it proposes a machine learning approach to detecting frame deletion. A number of discriminative features are extracted from the video bit stream and its reconstructed images. The features are based on prediction residuals, percentage of intra-coded macroblocks, quantization scales and reconstruction quality. The importance of these features is verified by using stepwise regression. Consequently, the dimensionality of the feature vectors is reduced using spectral regression where it is shown that the projected features of unaltered and forged videos are nearly separable. Machine learning techniques are used to report the true positive and false negative rates of the proposed solution. It is shown that the proposed solution works for detecting forged videos regardless of the number of deleted frames, as long as it is not a multiple of the length of a group of pictures. It is also shown that the proposed solution is applicable for the two modes of video compression, variable and constant bitrate coding.

Keywords: Digital forensics, frame deletion, video compression.

1. Introduction

Digital video evidence is captured through cameras that are deployed in streets, subways, train and underground stations, public schools, malls and the like. Digital video evidence can also be captured by users of cell phones who are acquainted with digital cameras and digital video recording. In all cases, the video is stored in a compressed lossy format that contains digital artifacts such as quantization and sampling.

Such abundance of compressed video material forms a potential piece of evidence in courtrooms. Because of inherent visual artifacts and possible fraud, courts usually call upon the testimony of forensics experts to subjectively assess the quality and authenticity of the digital video evidence. Augmenting or replacing this subjective assessment by an automated objective assessment helps increase the accuracy of deciding whether or not to admit the digital video as legal evidence. The automated objective assessment also speeds up the process of decision making.

There are a number of factors for the admissibility of compressed video as legal evidence in a courtroom. Factors of interest include video quality and video authenticity.

Assessing the quality of the compressed video in the absence of the reference can be performed on the whole video as one unit thus quantifying its quality [1]. The assessment can also be performed on image basis [2]. However in courtrooms, it makes more sense to quantify the quality of compressed video at a finer scale such as macroblock level assessment as reported in [3] and [4].

Likewise the authenticity of compressed video is examined. This is needed because compressed video can be manipulated through video editing, transcoding and translating. Prior to the admissibility of the compressed video to a courtroom it is of prime importance to examine the authenticity of the compressed video.

Research work on identifying tampering with compressed digital is reported in the literature. For instance, detecting double compression of MPEG-4 video is proposed in [5] using Markov-based features in the detection process. A double compression detection solution was also proposed in [6]. In their work, the length of the GoP used in the first encoding process is also estimated. Double compression of MPEG-2 video can also be detected by examining the distribution of quantized DCT coefficients as proposed in [7]. Additionally, periodic artifacts in the frequency spectrum of the distribution of reconstructed DCT coefficients can be used for detecting video transcoding as reported in [8].

Moreover, detecting forgery in digital video can be classified into spatial and temporal domains. Detection in the spatial domain refers to the existence of forgery fingerprints in a video frame. While detection in the temporal domain refers to the existence of such fingerprints across video frames.

Examples of forgery detection in the spatial domain include inspecting pixel authenticity to detect suspicious regions in a video of a static scene using noise characteristics [9]. The authors in [10] proposed detecting tampered regions in I-frames using a DCT-based watermark. Location of forged regions in a video can also be estimated using correlation of noise residuals. The correlation is modeled using a Gaussian Mixture Model followed by a Bayesian classifier [11].

On the other hand, detecting video forgery in the temporal domain mainly revolves around detecting frame deletion or insertion. For instance, motion-compensated edge artifacts can be used for detecting frame-based tampering such as deletion or addition of frames as reported in [12] and [13].

Noteworthy in the field of digital video forensics is the work carried out by Farid *et. al.* It has been shown that MPEG video sequences with double encoding introduce detectable static and temporal artifacts [14]. These artifacts are used to detect tampering of MPEG video sequences. Static artifacts are present when I-frames of an MPEG sequence are recompressed. And temporal artifacts are present as a result of frame deletion or insertion when frames move from one GOP to another.

More recently, a theoretical model of the forgery fingerprints that result from frame deletion or addition was developed [15]. This model was used to improve the video forensic techniques proposed techniques proposed by [14]. Additionally, game theory was used to analyze the interplay between a forensic investigator and a digital video forger. The analysis was used to identify the optimal actions of both the forensic investigator and the digital video forger.

The work presented in this paper proposes a machine learning approach for detecting frame deletion in forged videos. Briefly, feature vectors are extracted from video bit streams and decoded videos. The suitability of the selected features is verified by using stepwise regression. Consequently, the dimensionality of the features is reduced by means of spectral regression. Machine learning techniques are then used for detecting forged videos. The proposed solution can be used for initial inspection of videos under examination. It can also be used to complement other approaches such as visual inspection.

The paper is organized as follows. Section 2 reviews the most relevant existing solutions proposed by [14] and [15] whilst highlighting their advantages and limitations. Accordingly, the proposed work of this paper enhances upon the existing solutions. Section 3 illustrates an overview of the proposed solution. Section 4, presents the proposed features that will be used for detecting frame deletion. Section 5, presents the use of spectral regression with the proposed solution. The experimental results are presented in Section 6. Lastly, the paper is concluded in Section 7.

2. Existing work

One of the most significant solutions on frame deletion detection was originally proposed in [14]. Recently, [15] expanded the aforementioned work as mentioned in the introduction. Of interest is the proposal of an automatic approach to detecting frame deletion. Common to the work reported in [14] and [15], is that one feature was used to detect the existence of frame deletion. The feature is simply the energy of the prediction error or P-frames. The energy can be approximated by computing the absolute sum of prediction errors for all non-intra MBs within P-frames using the following equation:

$$e(n) = {}^{1}/_{N_i} \sum_i |P_i(n)| \qquad (1)$$

Where N_i is the number of non-intra coded MBs in each P-frame and $P_i(n)$ is the sum of absolute prediction error values for the n^{th} P-frame at MB index i.

The work reported in [14] made an important observation that due to frame deletion, periodic peaks are observed in the $e(n)$ sequence. It was proposed to manually inspect the sequence for such periodicity. Likewise, the discrete Fourier transformation of the sequence $E(k)$ = DFT{$e(n)$} can be inspected for peaks as well. It was reported that such peaks are more apparent if the number of frames deleted are a multiple of the sub-GoP length. For instance if a GoP structure of IBBPBBPBBPBB is used, the GoP length is 12 and the subgroup length is 3 (IBB or PBB). The peaks are most apparent when 3,6 or 9 frames are deleted. However, if a whole GoP is deleted or a multiple of GoPs are deleted then the peaks will not show in the $e(n)$ sequence.

The work reported in [15] formalized the aforementioned problem and proposed an automatic solution for the detection of peaks in the $e(n)$ sequence. It was also proposed to use a varying GoP length as an additional challenge in detecting frame deletion. In a nutshell, the $e(n)$ sequence is filtered by a median filter and compared against the non-filtered version. Let $d(n)$ denote the difference between the filtered and non-filtered sequences which is defined by:

$$d(n) = max(\, e(n) - \text{median}\{e(n-1), e(n), e(n+1)\}\,,\ 0) \qquad (2)$$

If a variable GoP length is used then this difference can then be thresholded to perform hypothesis testing where the null hypothesis, H_0, states that the video is not forged. The thresholding is expressed as follows:

$$decision = \begin{cases} H_0 \; if \; \mu_{|d(n)|} < Th \\ H_1 if \; \mu_{|d(n)|} \geq Th \end{cases} \qquad (3)$$

Where Th is a decision threshold which can be used to generate a Receiver Operation Curve (ROC).

In a similar fashion, if a fixed GoP length is used, then frame deletion is detected by measuring the strengths of the periodic peaks in $|DFT\{d(n)\}|$. It was reported that these peaks occur at N/T where N is the length of the $e(n)$ sequence and T is the number of P-frames in a GoP.

3. System overview

Existing video forensics detection solutions have their own limitations. Both of the reviewed solutions are considered landmarks in frame deletion detection, nonetheless they have a number of limitations. Namely, the work in [15] reported detection results for the case when the number of deleted frames is a multiple of sub-GoP length. Additionally, there is an assumption that frame deletion starts with either an I or a P frame. And the feature used of detecting the forgery is based on the prediction error of P-frames only. Lastly, both solutions made an implicit assumption that the forged videos are coded using Variable Bit Rate (VBR) coding not Constant Bit Rate (CBR) coding. Both solutions will not work with CBR coding. This is so because in CBR coding, the inefficiency in re-encoding the P-frames after forgery will not manifest itself in terms of higher prediction error only. Rather, it will more significantly be manifested in terms of higher percentage of intra coded MBs, lower PSNR quality and higher quantization scales.

Hence in this work, we expand upon the reviewed solutions by considering a new set of distinctive features. The proposed work is also suitable for both CBR and VBR coding. It can also detect frame deletion regardless of whether or not the number of deleted frames is a multiple of a sub-GoP length.

We propose a machine learning approach for detecting videos with deleted frames. Typically in machine learning, a detection system is trained with samples of unaltered and forged videos. A detection model is thereafter computed. The model can be applied to a video under examination to determine whether or not it is post-processed by means of frame deletion. More specifically, the detection system is trained with feature vectors extracted from various video bit streams. The feature extraction is elaborated upon in Section 4. Since the values stored in these feature vectors will vary in range, it is a good idea to normalize them. In this work we use z-scores for normalization. The mean and standard deviation vectors of the training feature vector set are stored and used for normalizing a feature vector for a video under examination. Additionally, in this work we propose the use of Spectral Regression (SR) for reducing the dimensionality of the feature vectors [17]. The integration of SR with the proposed solution is introduced in Section 5. The projection vectors that result from the SR are stored and reused for reducing the dimensionality of a feature vector representing a video under examination. The proposed frame deletion detection system is further illustrated in Figure 1.

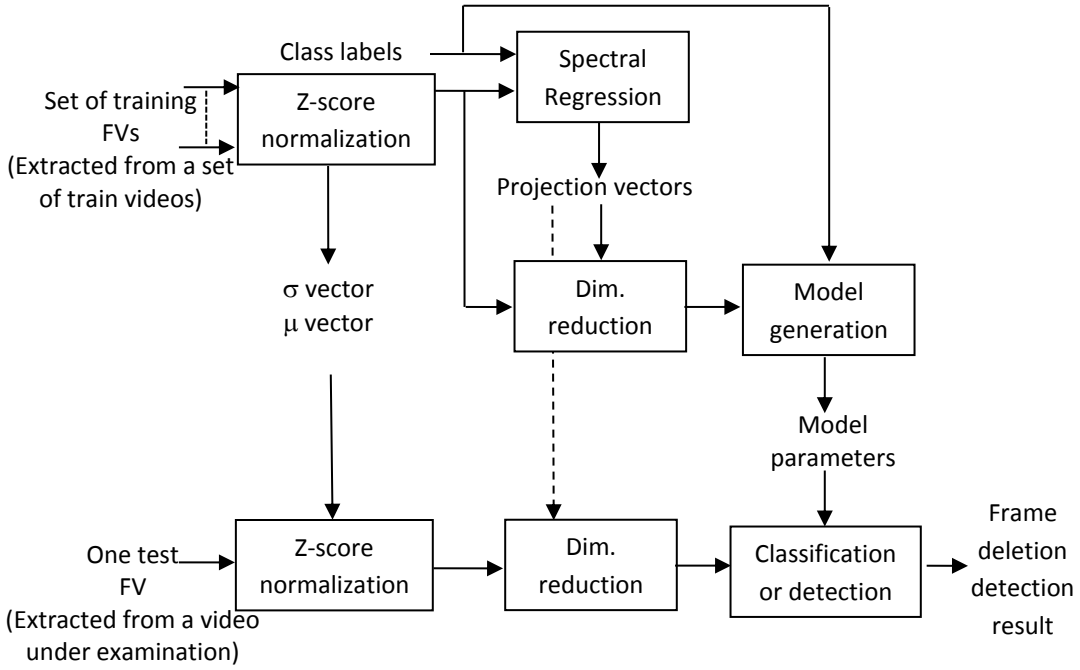


Figure 1. Block diagram of proposed frame deletion detection

In this work we use three machine learning techniques and compare the detection accuracy against existing work. The machine learning techniques used are the K Nearest Neighbor (KNN), logistic regression and Support Vector Machines (SVM).

4. Feature selection

This section proposes a new set of video features that will be used for detecting frame deletion. These features are extracted from the compressed video streams and decoded videos. The section will also introduce the use of stepwise regression for verifying the importance of the proposed features.

4.1 Proposed features

In this work, we propose to use eight features instead of one as reported in the reviewed work [15]. These features will be computed from both P and B frames. Each feature is computed over the whole video sequence for a particular frame type *i.e.* P or B frames. The list of features is summarized in Table 1.

Feature ID	Description
1, 2	The mean prediction residual energy of non-Intra coded MBs, μ_E The standard deviation of the prediction residual energy of non-Intra coded, σ_E
3, 4	The mean percentage of intra-coded MBs, μ_{Intra} The standard deviation of intra-coded MBs, σ_{Intra}
5, 6	The mean of estimated PSNR values, μ_{PSNR} The standard deviation of estimated PSNR values, σ_{PSNR}
7, 8	The mean of quantization scale values, μ_q The standard deviation of quantization scale values, σ_q

Table 1. List of features computed from P and B frames

The proposed features contain the mean and the standard deviation of the prediction residuals. It also contains an important feature which is based on the percentage of intra MBs.

The mean prediction residual energy of non-Intra coded MBs is computed using Equation 4:

$$\mu_E = {}^{1}/_{N} \sum_j \sum_i P_i(j), \quad i \in \{indices\ of\ non\ intra\ MBs\ at\ the\ jth\ frame\} \qquad (4)$$

Where *N* is the total number of predicted MBs in the video sequence for a P or a B frame. $P_i(j)$ is the sum of absolute residual values for the i^{th} MB at the j^{th} frame. The standard deviation of this feature is computed using Equation (5):

$$\sigma_E = \sqrt{E[(P_i(j) - \mu_E)^2]} \qquad (5)$$

Where the operator E denotes the expected value. The mean percentage of intra-coded MBs is computed using Equation (6):

$$\mu_{intra} = {}^{1}/_{N} \sum_j I(j) \qquad (6)$$

Where N is the total number of predicted P or B frames in a video sequence and $I(j)$ is the percentage of intra coded MBs in the j^{th} frame. The standard deviation of this feature is computed using Equation (7):

$$\sigma_{intra} = \sqrt{E[(I(j) - \mu_{intra})^2]} \qquad (7)$$

The mean of estimated PSNR values is computed using Equation (8):

$$\mu_{PSNR} = {}^{1}/_{N} \sum_j \hat{P}(j) \qquad (8)$$

Where N is the total number of predicted P or B frames in a video sequence and $\hat{P}(j)$ is the estimate of the PSNR of the j^{th} frame. The standard deviation of this feature is computed using Equation (9):

$$\sigma_{psnr} = \sqrt{E[(\hat{P}(j) - \mu_{psnr})^2]} \quad (9)$$

The mean of quantization scale values is computed using Equation (10):

$$\mu_q = {}^{1}/_{N} \sum_j \sum_i Q_i(j) \quad (10)$$

Where *N* is the total number of MBs in the video sequence for a P or a B frame. $Q_i(j)$ is the quantization scale of the i^{th} MB at the j^{th} frame. The standard deviation of this feature is computed using Equation (11):

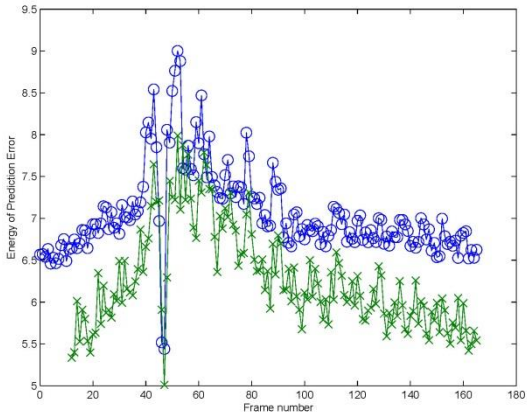
$$\sigma_q = \sqrt{E[(Q_i(j) - \mu_q)^2]} \quad (11)$$

It was observed that frame deletion does not affect the energy of prediction residuals only. Rather, it also affects the percentage of intra MBs. It also affects the variation of the percentage of intra MBs across frames. Additionally, frame deletion affects the quality of the decoded frames, hence the PSNR can be used as a feature as well. If CBR coding is used then frame deletion will also manifest itself in terms of the quantization scale. To accommodate higher prediction residuals in CBR coding, the coder tends to increase the quantization scale. Hence the average and the standard deviations of the quantization scale are needed as features.

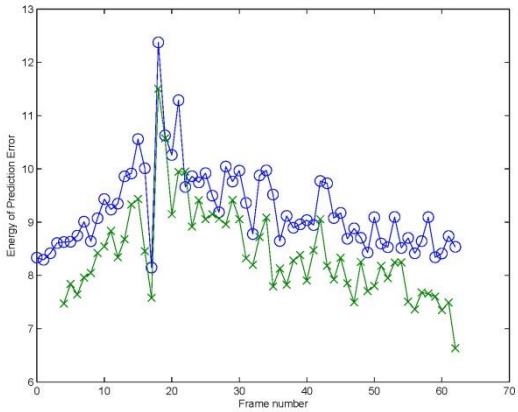
In a real life scenario, only the bit streams and the reconstructed videos are available. There is no access to the original raw video. Therefore, the PSNR of the decoded video streams can only be estimated. Such estimation is permissible and it is referred to 'no reference quality assessment'. To use the PSNR as a feature in this work we estimate it using the work proposed by the author in [3].

Figure 2 shows an example of the features extracted from unaltered and forged videos. In this example, the first 8 frames are deleted from the Coastguard sequence which was coded using VBR coding with a GoP structure of *IBBPBBPBBPBB*. The experimental setup and configuration is elaborated upon in the experimental results section. The figure shows the plots for both P and B frames for a number of features. For instance, part (a) and (b) of the figure show that the magnitude of the prediction residuals of the forged videos is generally less than that of the unaltered video. Note that the prediction error is restricted to non-intra MBs in P and B frames. Parts (c) and (d) of the figure show that the percentage of Intra MBs has increased in forged video. In some cases, it is also evident that the variation in this percentage is also higher for forged videos. Lastly, the PSNR plots are shown to follow a different pattern in the case of forged videos. Clearly, if CBR coding is used then the profile of the average quantization scale per frame will also play an important role in detecting frame deletion.

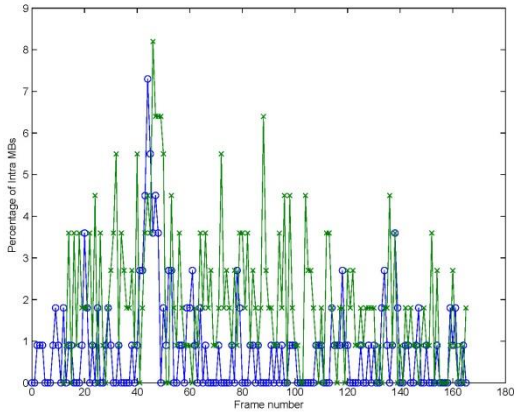
To further examine the choice of the features listed in Table 1, we propose the use of stepwise regression as introduced next.



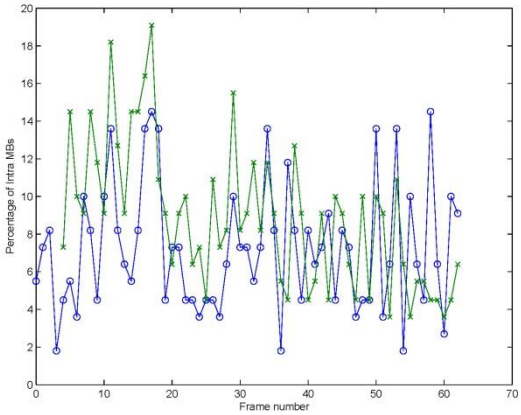
(a) B-frame's prediction energy profile



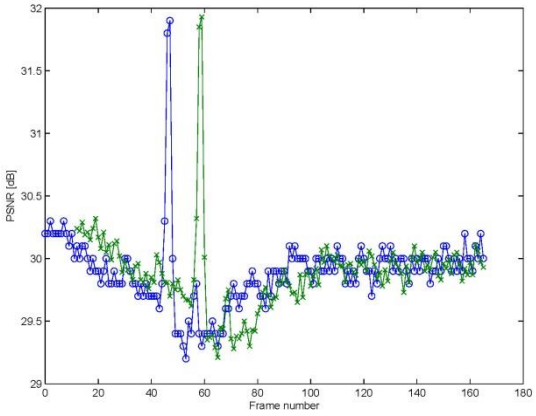
(b) P-frame's prediction energy profile



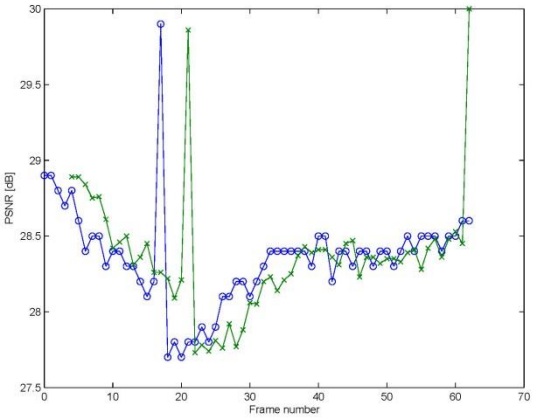
(c) B-frame's percentage of intra MBs profile



(d) P-frame's percentage of intra MBs profile



(e) B-frame's PSNR profile (using [3])



(f) B-frame's PSNR profile (using [3])

Figure 2. Example plots of discriminating features used for detecting frame deletion. The curves labeled with 'o' and 'x' represent unaltered and forged videos respectively.

4.2 Stepwise regression

Stepwise regression is an objective method of selecting important features. To apply stepwise regression to our problem, we treat the feature variables $x_1, x_2, \ldots, x_k$ as predictors where the k is the

number of features in each feature vector. Likewise, the class labels are treated as a response variable, y. In [16], the stepwise regression procedure is described using the following steps. In the first step, the procedure tests all possible one-predictor regression models in an attempt to find the predictor that has the highest correlation with the response variable. The model is of the form:

$$\hat{y} = \beta_0 + \beta_1 x_i \qquad (12)$$

A hypothesis test is conducted for each model where $H_0: \beta_1 = 0$ and $H_1: \beta_1 \neq 0$. The test is conducted using the well-known *T* test at a specific level of significance, say α = 0.1. The predictor that generates the largest absolute *T* value is selected. Refer to this predictor as x_1.

In the second step, the remaining *k-1* predictors are scanned for the best two-predictor regression model of the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_i \qquad (13)$$

This is achieved by testing all two-predictor models containing x_1 which was selected from the first step. The *T* value of the *k-1* models are computed for $H_0: \beta_2 = 0$. The predictor that generates the highest absolute *T* value is retained, Refer to this predictor as x_2.

Now that $\beta_2 x_2$ is added to the model, the procedure goes back and reexamines the suitability of including β_1 in the model. If the corresponding *T* value becomes insignificant (i.e. the alternative hypothesis H_1 is rejected.), x_1 is removed and the predictors are searched for a variable that generates the highest *T* value in the presence of $\beta_2 x_2$. In the third step, remaining *k-2* predictors are scanned for the best three-predictor regression model of the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_{2+} \beta_3 x_i \qquad (14)$$

And the procedure repeats until no further predictors are added or removed from the model.

Because stepwise regression is based on sample estimates, Type I and Type II errors are probable in which unimportant predictors are included in the model and important predictors are eliminated.

The stepwise regression procedure is applied to 36 video sequences. Two sets of videos are created. In one set the videos are compressed and remain unaltered. While in the other set, the videos are compressed, decompressed, forged in terms of frame deletion and compressed again.

The number of deleted frames range from 1 to 11. The details of the video sequences used and experimental set up are given in the experimental results section (Section 6). The features retained by stepwise regression for P and B frames are reported in Table 2.

No. Deleted frames	P-frames	B-frames
1	$\mu_E\ \mu_{PSNR}\ \sigma_{Intra}$	$\sigma_E\ \ \mu_E$
2	$\sigma_E\ \ \mu_E\ \sigma_{PSNR}\ \mu_{PSNR}\ \sigma_{Intra}$	$\sigma_E\ \mu_E\ \sigma_{Intra}$
3	$\mu_E\ \sigma_{Intra}$	$\mu_E\ \sigma_E\ \mu_{PSNR}$
4	$\sigma_E\ \mu_{PSNR}\ \sigma_{Intra}$	$\sigma_E\ \mu_E$
5	$\sigma_E\ \ \mu_E\ \sigma_{Intra}$	$\sigma_E\ \ \mu_E\ \mu_{PSNR}\ \sigma_{Intra}$
6	$\mu_E\ \sigma_{Intra}$	$\sigma_E\ \mu_E\ \mu_{PSNR}$
7	$\mu_E\ \mu_E\ \sigma_{Intra}$	$\sigma_E\ \ \mu_E\ \mu_{PSNR}$
8	$\sigma_{Intra}\ \ \mu_E\ \sigma_{Intra}$	$\sigma_E\ \mu_E\ \ \mu_{PSNR}\ \sigma_{Intra}$
9	$\mu_E\ \mu_{PSNR}\ \sigma_{Intra}$	$\sigma_E\ \ \mu_E$
10	$\mu_E\ \mu_{PSNR}\ \sigma_{Intra}$	$\sigma_E\ \mu_E$
11	$\sigma_E\ \mu_E\ \sigma_{PSNR}\ \ \mu_{PSNR}\ \sigma_{Intra}$	$\sigma_E\ \ \mu_E\ \sigma_{Intra}$

Table 2. Features retained after applying stepwise regression for P and B frames

It is shown from the table that various combinations of features are retained for the detection of different numbers of deleted frames. It is interesting to see that in few cases, the stepwise regression procedure retained the features pertaining to the energy of the prediction residual on its own. This is consistent with what is reported in [14] and [15]. However, for many other cases apart from deleting 1,4,9 or 10 frames, the features based on the percentage of Intra MBs and PSNR are also retained by the stepwise regression procedure.

5. Dimensionality reduction

As mentioned in the system's overview, in this work we reduce the dimensionality of the feature vectors prior to classification using Spectral Regression. One of the attractive features of spectral regression is that it reduces the dimensionality to the total number of classification classes minus 1. This section starts be reviewing spectral regression. We then apply it to one of the video test sequences and show that the resultant features are reasonably separable.

Spectral Regression Discriminant Analysis (SRDA) has significant computational advantage over Linear Discriminant Analysis (LDA). It is shown that by linking LDA and classical regression, a LDA solution can be computed by solving a set of linear equations as originally proposed by [17]. It was proposed to combine spectral graph analysis and regression to provide an effective approach for discriminant analysis. For completeness, the algorithm is summarized in this subsection.

For a set of *m* feature vectors $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m$, belonging to *c* classes, the objective function of LDA is as follows:

$$a^* = \arg_a \max \frac{a^T S_b a}{a^T S_w a} \qquad (15)$$

where S_w is the within-class scatter matrix and S_b is the between-class scatter matrix.

In order to solve the LDA eigen-problem in Equation 7 efficiently, the following theorem is used.

Let $\bar{y}$ be the eigenvector of eigen-problem, $W\bar{y} = \lambda\bar{y}$ with eigenvalue λ. If $\bar{X}^T a = \bar{y}$ then 'a' is the eigenvector of eigen-problem in Equation 7 with the same eigenvalue λ. Where $\bar{X}$ is the centered data matrix, $W(k)$ is a $m_k \times m_k$ matrix with all elements equal to $(1/m_k)$, m_k is the number of data points in k^{th} class, m is the number of total training data points, n is the number of features, c is the number of classes. The LDA basis functions can be obtained by solving the eigen-problem to get $\bar{y}$ then finding 'a' which satisfies $\bar{X}^T a = \bar{y}$. A possible way is to find 'a' is:

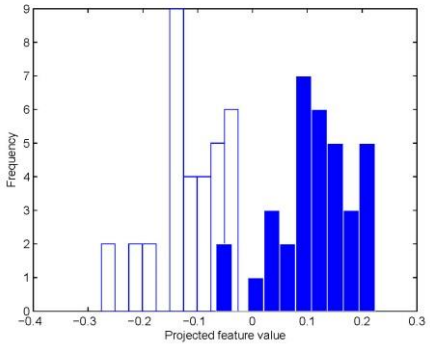
$$a = \arg_a \min \sum_{i=1}^{m} (a^T \bar{x}i - \bar{y}i)^2 + \propto \|a\|^2 \qquad (16)$$

Where $\bar{y}i$ is the i^{th} element of $\bar{y}$.

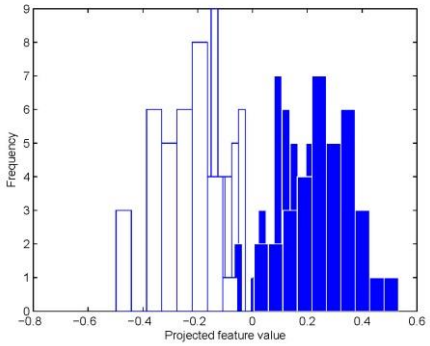
In [17] it is shown that for a large number of features, this technique produces more stable and meaningful solutions in comparison to Linear Discriminant Analysis (LDA) and its extensions like regularized LDA, uncorrelated LDA and LDA with QR decomposition. The name spectral regression Discriminant Analysis follows from the fact that a two-step approach is followed in which spectral analysis of the graph matrix **W** and regression are combined.

Having normalized the feature vectors and applied spectral regression, the number of features per class will be reduced to one. That is, the number of feature variables for the unaltered and the forged videos after applying spectral regression will be reduced from 8 (as listed in Table 1) to one feature variable only. Hence the histograms of the feature vectors of both classes can be visualized as shown in Figure 3.

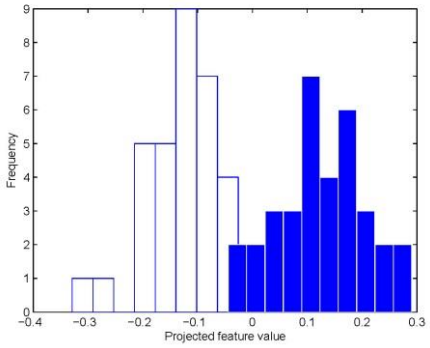
In each histogram, the non-shaded bins represent the projected feature vectors of the original unaltered videos and the shaded bins represent the projected feature vectors of the forged video.



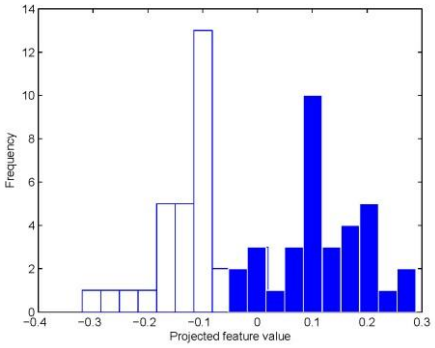
(a) 1 deleted frame



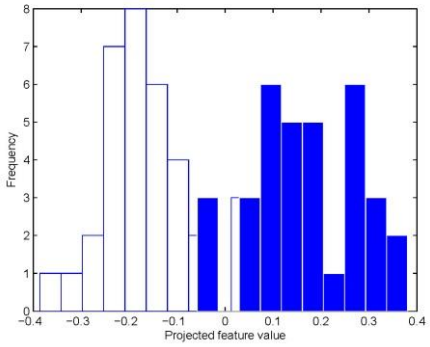
(b) 2 deleted frames



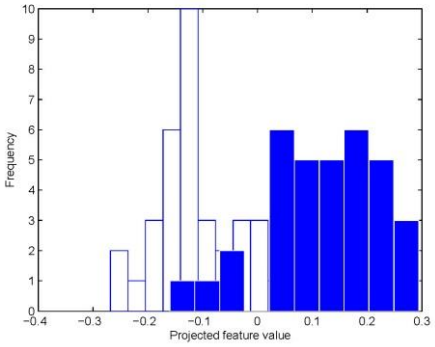
(c) 3 deleted frames



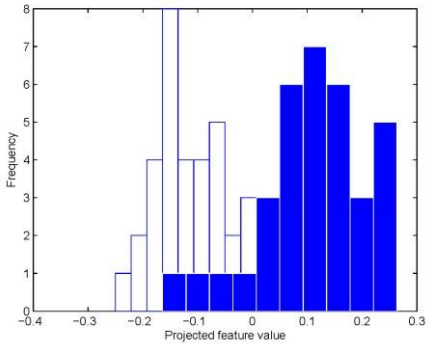
(d) 4 deleted frames



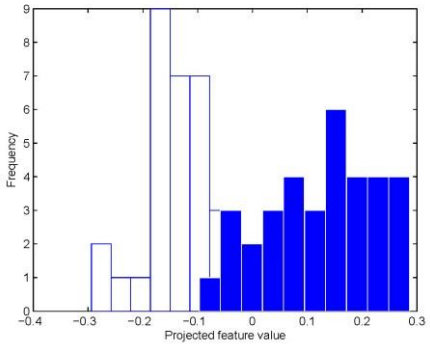
(e) 5 deleted frames



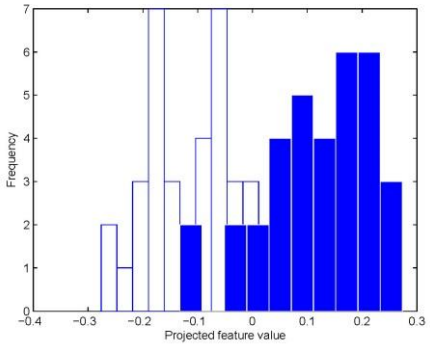
(f) 6 deleted frames



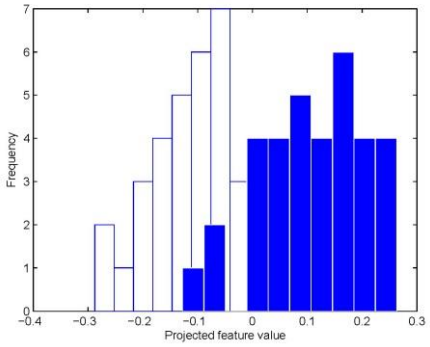
(g) 7 deleted frames



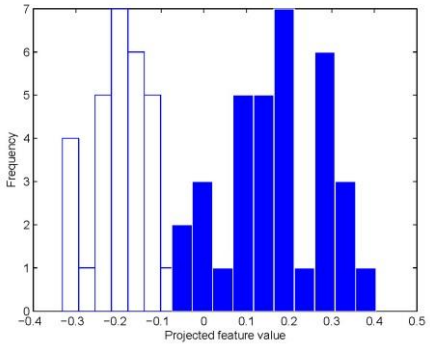
(h) 8 deleted frames



(i) 9 deleted frames



(j) 10 deleted frames



(k) 11 deleted frames

Figure 3. Histogram of projected features using spectral regression.

The figure plots the histograms for 1 to 11 deleted frames. The x-axis represents the value of the projected feature vectors using spectral regression and the y-axis represents the count/frequency of these values. It is shown that spectral regression resulted in nearly separable classes. It is shown in the experimental results section that combining spectral regression with a classification technique like Support Vector Machines, results in very accurate forgery detection. The histograms are generated for 36 test sequences. The test sequences and the coding configurations are listed in the experimental results section.

6. Experimental results

This section evaluates the performance of the proposed frame deletion detection technique. For a fair comparison with existing solutions, we use a similar experimental setup to that reported in [15]. The compression parameters are typical and are reused from the work reported in [15]. An MPEG-2 codec which is an implementation of the ISO/IEC DIS 13818-2 international standard [18] is used to compress 36 standard QCIF test sequences using VBR and CBR coding. The quantizer triplet for the VBR coder is {20, 16 and 12} for I, P and B frames respectively. The CBR coding on the other hand, used a constant bitrate of 250 kbit/s. The GoP structure is N=12 and M=3, that is *IBBPBBPBBPBB.* For variable GoP structure coding, and following the experimental setup of [15], the length of each GoP was randomly

assigned to either N=12 or N=15. The test sequences are: Akiyo, Bowing, Bridge-Close, Bridge-Far, Carphone, City, Claire, Coastguard, Container, Crew, Deadline, Flower Garden, Football, Foreman, Galleon, Grandma, Hall, Harbour, Highway, Husky, Intros, Pamphlet, Mobile, Mother and Daughter, News, Paris, Salesman, Sign-Irene, Silent, Soccer, stefan, Students, Table, Tempete, Vtc1nw and WashDC.

Four unaltered sets of videos are compressed using VBR and CBR coding each with fixed and varying GoP lengths. Following the work reported in [14] and [15], the forged videos are created by decoding the mentioned set, deleting the first one or more frames up to a GoP length and re-encoding using the same compression parameters of the original videos.

In the following experiments, the true positive (TP) rates and the false alarm rates (FA) are reported using KNN (with K set to 1 with an Euclidean distance similarity measure), logistic regression and SVM with a quadratic kernel. In each run of the experiment, 35 test sequences are used for training and one sequence is used for testing. Thus, 36 results are obtained. Note that the test video sequence is not part of the training data, hence, the model is sequence independent. For readability purposes, the average true positive and false negative rates of the 36 results are reported for detecting the deletion of 1 or more frames up to a GoP length.

The results in Table 3 show the TP and FA rates for detecting the deletion of video frames. The results are reported for both VBR and CBR videos using a fixed GoP length of 12.

	CBR coding						VBR coding					
	KNN		L. Reg.		SVM		KNN		L. Reg.		SVM	
No. frames	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA
1	91	3	94	6	94	3	94	9	100	6	100	9
2	94	3	94	0	91	0	97	3	97	3	97	0
3	91	3	94	9	94	9	94	12	97	6	94	6
4	100	0	100	0	97	0	88	15	91	9	91	9
5	100	3	91	6	94	6	94	3	91	9	91	9
6	97	6	94	6	94	12	91	15	88	12	94	12
7	97	6	97	0	97	0	82	12	91	9	97	15
8	94	3	97	3	97	3	88	12	88	12	88	12
9	97	9	91	3	88	3	82	15	94	12	97	12
10	91	12	91	3	91	3	79	12	91	9	91	12
11	100	0	100	6	100	6	94	0	91	6	91	6
Avg	95.6	4.4	94.8	3.8	94.3	4.1	89.4	9.8	92.6	8.5	93.7	9.3

Table 3. True positive and false alarm rates (%) for frame deletion using a fixed GoP length.

The first column in the table indicates the number of deleted frames. The first set of results pertains to CBR coding and the second set pertains to VBR coding. It is shown that frame deletion detection using the proposed technique works regardless of the number of deleted frames. For instance, if 4,6 or 8 frames are deleted then the true positive and false negative rates are similar. Hence the restriction of detecting forged videos with a number of deleted frames being a multiple of sub-GoP lengths (i.e. 3, 6 and 9) is eliminated. It is also observed that the proposed technique does not necessarily perform better at detecting forged videos with a number of deleted frames equal to multiples of sub-GoP lengths. For example, the true positive and false negatives for detecting 7 deleted frames are more accurate than detecting 6 deleted frames. As mentioned previously, this is one of the restrictions in the existing solutions reported in [14] and [15]. The set of reported results in Table 3 also indicate that the proposed solution works for detecting deleted frames of videos using both VBR and CBR coding. This is evident by examining the average true positive and average false negative rates for both approaches as reported in the last row of the table. Thus, the restriction of detecting frame deletion using VBR coding only is also eliminated. Again, this was another restriction in the reported solutions of [14] and [15]. Lastly, the average results show that the three classifiers performed well in terms of average true positive and false alarm rates.

For a comparison with the reviewed work reported in [15], the averages of true positive and false alarm rates for detecting the deletion of 3,6 and 9 frames are reported in Table 4. Again the first column reports the number of deleted frames.

	VBR coding							
	KNN		L. Reg.		SVM		Reviewed	
No. frames	TP	FA	TP	FA	TP	FA	TP	FA
3	94	12	97	6	94	6	84	10
6	91	15	88	12	94	12	90	10
9	82	15	94	12	97	12	97	10
Avg	89.0	14.0	93.0	10.0	95.0	10.0	90.3	10.0

Table 4. True positive and false alarm rates (%) for frame deletion using a fixed GoP length. Comparison with existing work for deletion detection of 3, 6 and 9 frames.

It is shown from the table that on average the true positive rates using the proposed solution with SMV classification are higher than the reviewed work by around 5%. Namely, the average true positive rate using the proposed SVM solution is 95% and that of the reviewed work is 90.3%. The false alarm rates on the other hand are at par in both cases. Again, the proposed solution has the advantage of detecting

forged videos regardless of the number of deleted frames and regardless of the coding mode (VBR versus CBR).

The same set of experiments reported in Table 3 are repeated using a varying GoP length. The results are reported in Table 5.

	CBR coding						VBR coding					
	KNN		L. Reg.		SVM		KNN		L. Reg.		SVM	
No. frames	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA
1	91	9	94	6	91	6	82	12	94	9	97	12
2	94	6	97	3	97	3	100	0	100	3	100	3
3	88	15	94	6	91	3	91	12	85	12	91	12
4	94	6	91	3	91	3	91	9	91	9	94	15
5	97	6	97	3	97	6	97	0	97	0	97	0
6	94	6	94	3	94	3	88	15	91	9	94	12
7	97	6	97	3	97	3	82	9	91	9	97	9
8	94	12	94	9	94	9	100	3	94	3	94	3
9	88	6	88	12	94	12	85	12	91	12	94	15
10	91	12	94	9	94	9	94	9	94	9	94	9
11	94	3	94	3	94	3	97	0	97	0	97	0
Avg	92.9	7.9	94.0	5.5	94.0	5.5	91.5	7.4	93.2	6.8	95.4	8.2

Table 5. True positive and false alarm rates (%) for frame deletion using a varying GoP length.

Comparing the results of Table 5 to Table 3, it is shown that proposed technique is robust to change in the GoP length. On average, the results are more or less the same. In [15], this experiment was reported for deleting 6 frames only. Hence, in Table 6 we show a comparison between the proposed technique and the reviewed one for deleting 6 frames using a varying GoP length.

	VBR coding							
	KNN		L. Reg.		SVM		Reviewed	
No. frames	TP	FA	TP	FA	TP	FA	TP	FA
6	88	15	91	9	94	12	78	9

Table 6. True positive and false alarm rates for frame deletion using a varying GoP length. Comparison with existing work for deletion detection of 6 frames.

The results in the table indicate that using logistic regression or SVM as a classification tool in the proposed technique outperforms the reviewed work. For instance, using logistic regression, the true positive rate is 91% and the false alarm rate it 9%. Whereas in the reviewed work, the rates are 78% and 9% respectively.

To quantify the proposed solution in terms of accuracy and consistency, the averages and standard deviations of the classification rates are reported in Table 7. As the case with Tables 3 and 5, the experiment is repeated with a varying number of deleted frames ranging from 1 to 11.

	KNN		LR		SVM	
No. frames	Classification rate	standard deviation	Classification rate %	standard deviation	Classification rate	standard deviation
1	0.94	0.2	0.94	0.16	0.96	0.14
2	0.96	0.14	0.97	0.12	0.96	0.14
3	0.94	0.16	0.93	0.18	0.94	0.16
4	1.00	0	1.00	0	0.96	0.14
5	0.99	0.09	0.93	0.18	0.97	0.12
6	0.96	0.14	0.94	0.16	0.93	0.18
7	0.96	0.14	0.99	0.09	0.99	0.09
8	0.96	0.14	0.97	0.12	0.99	0.09
9	0.94	0.16	0.94	0.16	0.93	0.18
10	0.90	0.21	0.94	0.16	0.94	0.16
11	1.00	0	0.97	0.12	0.97	0.12
Avg	0.96	0.13	0.96	0.13	0.96	0.14

Table 7. Average and standard deviation of classification rates of the proposed solution, CBR coding used.

It is shown that the standard deviation ranges from 0 to 0.18 using the classification techniques of logistic regression and SVM. The average classification rate ranges from 0.94 to 1.0. This gives a good indication that the proposed solution is both accurate and reasonably consistent.

Lastly, it is worth mentioning that the original video might be re-encoded without frame deletion. It is important to be able to distinguish between re-encoded videos with or without frame deletion. In Table 8, we show that the proposed solution can distinguish between re-encoding videos with and without frame deletion.

	CBR coding						VBR coding					
	KNN		L. Reg.		SVM		KNN		L. Reg.		SVM	
Frame num.	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA
1	94	6	94	6	94	6	74	21	79	15	79	15
2	94	3	97	3	97	3	76	18	82	15	79	12
3	94	9	94	6	94	6	76	18	71	21	68	12
4	94	6	97	3	97	3	85	15	91	9	91	9
5	88	9	94	6	100	9	68	24	82	18	85	18
6	91	9	97	9	100	9	65	32	82	15	76	15
7	94	3	97	6	97	6	91	6	91	6	91	6
8	88	6	91	9	95	9	76	12	82	12	79	12
9	94	6	94	3	94	0	68	29	76	26	74	12
10	94	3	97	3	94	0	88	21	85	9	85	9
11	96	8	94	2	94	1	90	18	86	8	88	6
Avg	92.8	6.2	95.1	5.1	96.0	4.7	77.9	19.5	82.5	14.0	81.4	11.5

Table 8. True positive and false alarm rates for frame deletion using a fixed GoP length. Both the forged and non-forged videos are encoded.

The results in the table can be compared against the results of Table 3. It is shown that the former results are less accurate. This is understood as distinguishing between re-encoded videos with and without frame deletion is more challenging than distinguishing between and original video and a re-encoded video with frame deletion.

7. Conclusion

A machine learning approach is proposed for detecting frame deletion in digital video. Features are extracted from the bit stream and the reconstructed images of videos under examination. The feature set is based on the prediction residuals, percentage of intra-coded macroblocks, quantization scales and an estimate of the PSNR values. Stepwise regression was used to verify the importance of the aforementioned features. The selection of features was shown to be suitable for both VBR and CBR modes of coding. Additionally, in the proposed classification system, the dimensionality of the features is reduced using spectral regression. A number of machine learning techniques were then used to detect frame deletion. The used techniques are KNN, logistic regression and SVMs. The detection accuracy was assessed by reporting the true positive and false negative rates. The experimental results showed that the proposed system is capable of detecting forged videos with various numbers of deleted frames. The system is also as accurate when a varying length GoP is employed. On average, a true positive rate of around 95% and a false negative rate of 4% were reported. The proposed system had a clear advantage

over existing solutions in terms of accuracy and flexibility. In future work, the proposed solution can be modified and extended in order to determine the exact location of the deleted frames and not just detect the existence of frame deletion.

References:

[1] L. Yu-xin, K. Ragip and B. Udit, "Video classification for video quality prediction," Journal of Zhejiang University Science A, 7(5), pp. 919-926, 2006.

[2] A. Ichigaya, M. Kurozumi, N. Hara, Y. Nishida, and E. Nakasu, "A method of estimating coding PSNR using quantized DCT coefficients", IEEE Transactions on Circuits and Systems for Video Technology, 16(2), pp. 251–259, February 2006.

[3] T. Shanableh, "No-Reference PSNR Identification of MPEG Video Using Spectral Regression and Reduced Model Polynomial Networks," IEEE Signal Processing Letters, 17(8), August, 2010

[4] T. Shanableh, "Prediction of Structural Similarity Index of Compressed Video at a Macroblock Level," IEEE Signal Processing Letters, 18(5), May, 2011.

[5] X. Jiang, W. Wang, T. Sun; Y. Q. Shi and S. Wang, "Detection of Double Compression in MPEG-4 Videos Based on Markov Statistics," Signal Processing Letters, IEEE , 20(5), pp.447,450, May 2013.

[6] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesana, A. Piva and M. Barni, "Detection of video double encoding with GOP size estimation," 2012 IEEE International Workshop on Information Forensics and Security (WIFS), , pp.151,156, 2-5 Dec. 2012

[7] Y. Su and J. Xu, "Detection of Double-Compression in MPEG-2 Videos," 2nd International Workshop on Intelligent Systems and Applications (ISA), pp.1,4, 22-23 May 2010

[8] J. Xu, Y. Su and X. You, "Detection of video transcoding for digital forensics," 2012 International Conference on Audio, Language and Image Processing (ICALIP), pp.160,164, July 2012

[9] M. Kobayashi, T. Okabe and Y. Sato, "Detecting Forgery From Static-Scene Video Based on Inconsistency in Noise Level Functions," Information Forensics and Security, IEEE Transactions on , 5(4), pp.883,892, Dec. 2010

[10] Y. Zhou, F.-Z. Zeng and G.-F. Yang, "The research for tamper forensics on MPEG-2 video based on compressed sensing," International Conference on Machine Learning and Cybernetics (ICMLC), pp.1080,1084, 15-17 July 2012

[11] C. Hsu, T. Hung, C.-W. Lin and C. Hsu, "Video forgery detection using correlation of noise residue," IEEE 10th Workshop on Multimedia Signal Processing, 2008 , pp.170,174, 8-10 Oct. 2008

[12] Q. Dong, G. Yang and N. Zhu “A MCEA based passive forensics scheme for detecting frame-based video tampering, Digital Investigation, 9(2), pp. 151-159, November 2012

[13] Y. Su, J. Zhang and J. Liu, "Exposing Digital Video Forgery by Detecting Motion-Compensated Edge Artifact," International Conference on Computational Intelligence and Software Engineering, CiSE 2009, pp.1,4, 11-13 December 2009

[14] W. Wang and H. Farid, “Exposing digital forgeries in video by detecting double MPEG compression,” in Proc. ACM Multimedia and Security Workshop, Geneva, Switzerland, pp. 37–47, 2006

[15] M. Stamm, W. S. Lin and K. J. Ray Liu, “Temporal Forensics and Anti-Forensics for Motion Compensated Video,” IEEE Transactions on Information Forensics And Security, 7(4), August 2012.

[16] W. Mendenhall and T. Sincich, Statistics for Engineering and Sciences, 5th edition, Pearson, 2007.

[17] D. Cai, X. He, and J. Han, "SRDA: An Efficient Algorithm for Large Scale Discriminant Analysis", IEEE Transactions on Knowledge and Data Engineering, 20(1), pp. 1-12, 2008

[18] Implementation of the ISO/IEC DIS 13818-2, available online, http://www.mpeg.org/MSSG/.

Tamer Shanableh earned his Ph.D. in Electronic Systems Engineering in 2002 from the University of Essex, UK. From 1998 to 2001, he was a senior research officer at the University of Essex, during which, he collaborated with BTexact on inventing video transcoders. He joined Motorola UK Research Labs in 2001. During his affiliation with Motorola, he contributed to establishing a new profile within the ISO/IEC MPEG-4 known as the Error Resilient Simple Scalable Profile. He joined the American University of Sharjah in 2002 and is currently an associate professor of computer science. Dr. Shanableh spent the summers of 2003, 2004, 2006, 2007 and 2008 as a visiting professor at Motorola multimedia Labs. He spend the spring semester of 2012 as a visiting academic at the Multimedia and Computer Vision and Lab at the School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, U.K . His research interests include digital video processing and pattern recognition.